

# File I

## Implementation

### 1 l3backend-basics Implementation

```
1 <*package>
```

Whilst there is a reasonable amount of code overlap between backends, it is much clearer to have the blocks more-or-less separated than run in together and DocStripped out in parts. As such, most of the following is set up on a per-backend basis, though there is some common code (again given in blocks not interspersed with other material).

All the file identifiers are up-front so that they come out in the right place in the files.

```
2 \ProvidesExplFile
3 <*dvipdfmx>
4   {l3backend-dvipdfmx.def}{2021-01-09}{ }
5   {L3 backend support: dvipdfmx}
6 </dvipdfmx>
7 <*dvips>
8   {l3backend-dvips.def}{2021-01-09}{ }
9   {L3 backend support: dvips}
10 </dvips>
11 <*dvisvgm>
12   {l3backend-dvisvgm.def}{2021-01-09}{ }
13   {L3 backend support: dvisvgm}
14 </dvisvgm>
15 <*luatex>
16   {l3backend-luatex.def}{2021-01-09}{ }
17   {L3 backend support: PDF output (LuaTeX)}
18 </luatex>
19 <*pdftex>
20   {l3backend-pdftex.def}{2021-01-09}{ }
21   {L3 backend support: PDF output (pdfTeX)}
22 </pdftex>
23 <*xetex>
24   {l3backend-xetex.def}{2021-01-09}{ }
25   {L3 backend support: XeTeX}
26 </xetex>
```

Check if the loaded kernel is at least enough to load this file. The kernel date has to be at least equal to \ExplBackendFileDate or later. If \\_kernel\_dependency\\_version\\_check:Nn doesn't exist we're loading in an older kernel, so it's an error anyway. With time, this test should vanish and only the dependency check should remain.

```
27 \cs_if_exist:NTF \_kernel_dependency_version_check:Nn
28   {
29     \_kernel_dependency_version_check:Nn {2020-09-01}
30     <dvipdfmx>    {l3backend-dvipdfmx.def}
31     <dvips>       {l3backend-dvips.def}
32     <dvisvgm>     {l3backend-dvisvgm.def}
33     <luatex>      {l3backend-luatex.def}
34     <pdftex>      {l3backend-pdftex.def}
35     <xetex>       {l3backend-xetex.def}
```

```

36 }
37 {
38   \cs_if_exist_use:cF { @latex@error } { \errmessage }
39   {
40     Mismatched-LaTeX-support-files-detected. \MessageBreak
41     Loading~aborted!
42   }
43   { \use:c { @ehd } }
44   \tex_endinput:D
45 }

```

The order of the backend code here is such that we get somewhat logical outcomes in terms of code sharing whilst keeping things readable. (Trying to mix all of the code by concept is almost unmanageable.) The key parts which are shared are

- Color support is either dvips-like or LuaTeX/pdfTeX-like.
- LuaTeX/pdfTeX and dvipdfmx/X<sub>Y</sub>TeX share drawing routines.
- X<sub>Y</sub>TeX is the same as dvipdfmx other than image size extraction so takes most of the same code.

`\__kernel_backend_literal:e` The one shared function for all backends is access to the basic `\special` primitive: it has slightly odd expansion behaviour so a wrapper is provided.

```

\__kernel_backend_literal:n
\__kernel_backend_literal:x
46 \cs_new_eq:NN \__kernel_backend_literal:e \tex_special:D
47 \cs_new_protected:Npn \__kernel_backend_literal:n #1
48 { \__kernel_backend_literal:e { \exp_not:n {#1} } }
49 \cs_generate_variant:Nn \__kernel_backend_literal:n { x }

```

(End definition for `\__kernel_backend_literal:e`.)

## 1.1 dvips backend

```

50 <dvips>

```

`\__kernel_backend_literal_postscript:n` Literal PostScript can be included using a few low-level formats. Here, we use the form with no positioning; this is overall more convenient as a wrapper. Note that this does require that where position is important, an appropriate wrapper is included.

```

51 \cs_new_protected:Npn \__kernel_backend_literal_postscript:n #1
52 { \__kernel_backend_literal:n { ps:: #1 } }
53 \cs_generate_variant:Nn \__kernel_backend_literal_postscript:n { x }

```

(End definition for `\__kernel_backend_literal_postscript:n`.)

`\__kernel_backend_postscript:n` PostScript data that does have positioning, and also applying a shift to `SDict` (which is not done automatically by `ps:` or `ps::`, in contrast to `!` or `"`).

```

54 \cs_new_protected:Npn \__kernel_backend_postscript:n #1
55 { \__kernel_backend_literal:n { ps: SDict ~ begin ~ #1 ~ end } }
56 \cs_generate_variant:Nn \__kernel_backend_postscript:n { x }

```

(End definition for `\__kernel_backend_postscript:n`.)

PostScript for the header: a small saving but makes the code clearer. This is held until the start of shipout such that a document with no actual output does not write anything.

```

57 \bool_if:NT \g__kernel_backend_header_bool
58 {

```

```

59 \cs_if_exist:NTF \AtBeginDvi
60 { \AtBeginDvi }
61 { \use:n }
62 { \__kernel_backend_literal:n { header = l3backend-dvips.pro } }
63 }

```

`\__kernel_backend_align_begin:` In **dvips** there is no built-in saving of the current position, and so some additional PostScript is required to set up the transformation matrix and also to restore it afterwards. Notice the use of the stack to save the current position “up front” and to move back to it at the end of the process. Notice that the `[begin]/[end]` pair here mean that we can use a run of PostScript statements in separate lines: not *required* but does make the code and output more clear.

```

64 \cs_new_protected:Npn \__kernel_backend_align_begin:
65 {
66   \__kernel_backend_literal:n { ps::[begin] }
67   \__kernel_backend_literal_postscript:n { currentpoint }
68   \__kernel_backend_literal_postscript:n { currentpoint~translate }
69 }
70 \cs_new_protected:Npn \__kernel_backend_align_end:
71 {
72   \__kernel_backend_literal_postscript:n { neg-exch-neg-exch~translate }
73   \__kernel_backend_literal:n { ps::[end] }
74 }

```

(End definition for `\__kernel_backend_align_begin:` and `\__kernel_backend_align_end:.`)

`\__kernel_backend_scope_begin:` Saving/restoring scope for general operations needs to be done with **dvips** positioning (try without to see this!). Thus we need the `ps:` version of the special here. As only the graphics state is ever altered within this pairing, we use the lower-cost `g`-versions.

```

75 \cs_new_protected:Npn \__kernel_backend_scope_begin:
76 { \__kernel_backend_literal:n { ps:gsave } }
77 \cs_new_protected:Npn \__kernel_backend_scope_end:
78 { \__kernel_backend_literal:n { ps:grestore } }

```

(End definition for `\__kernel_backend_scope_begin:` and `\__kernel_backend_scope_end:.`)

```

79 </dvips>

```

## 1.2 LuaTeX and pdfTeX backends

```

80 <*luatex | pdftex>

```

Both LuaTeX and pdfTeX write PDFs directly rather than via an intermediate file. Although there are similarities, the move of LuaTeX to have more code in Lua means we create two independent files using shared DocStrip code.

`\__kernel_backend_literal_pdf:n` This is equivalent to `\special{pdf:}` but the engine can track it. Without the `direct` keyword everything is kept in sync: the transformation matrix is set to the current point automatically. Note that this is still inside the text (BT ...ET block).

`\__kernel_backend_literal_pdf:x`

```

81 \cs_new_protected:Npn \__kernel_backend_literal_pdf:n #1
82 {
83   <*luatex>
84   \tex_pdfextension:D literal
85   </luatex>
86   <*pdftex>

```

```

87     \tex_pdfliteral:D
88   </pdftex>
89     { \exp_not:n {#1} }
90   }
91   \cs_generate_variant:Nn \__kernel_backend_literal_pdf:n { x }

```

(End definition for \\_\_kernel\_backend\_literal\_pdf:n.)

\\_\_kernel\_backend\_literal\_page:n Page literals are pretty simple. To avoid an expansion, we write out by hand.

```

92   \cs_new_protected:Npn \__kernel_backend_literal_page:n #1
93   {
94     <*luatex>
95       \tex_pdfextension:D literal ~
96     </luatex>
97     <*pdftex>
98       \tex_pdfliteral:D
99     </pdftex>
100       page { \exp_not:n {#1} }
101   }

```

(End definition for \\_\_kernel\_backend\_literal\_page:n.)

\\_\_kernel\_backend\_scope\_begin: Higher-level interfaces for saving and restoring the graphic state.

```

\__kernel_backend_scope_end:
102 \cs_new_protected:Npn \__kernel_backend_scope_begin:
103 {
104 <*luatex>
105   \tex_pdfextension:D save \scan_stop:
106 </luatex>
107 <*pdftex>
108   \tex_pdfsave:D
109 </pdftex>
110 }
111 \cs_new_protected:Npn \__kernel_backend_scope_end:
112 {
113 <*luatex>
114   \tex_pdfextension:D restore \scan_stop:
115 </luatex>
116 <*pdftex>
117   \tex_pdfrestore:D
118 </pdftex>
119 }

```

(End definition for \\_\_kernel\_backend\_scope\_begin: and \\_\_kernel\_backend\_scope\_end:.)

\\_\_kernel\_backend\_matrix:n Here the appropriate function is set up to insert an affine matrix into the PDF. With pdfTeX and LuaTeX in direct PDF output mode there is a primitive for this, which only needs the rotation/scaling/skew part.

```

120 \cs_new_protected:Npn \__kernel_backend_matrix:n #1
121 {
122 <*luatex>
123   \tex_pdfextension:D setmatrix
124 </luatex>
125 <*pdftex>
126   \tex_pdfsetmatrix:D
127 </pdftex>

```

```

128         { \exp_not:n {#1} }
129     }
130 \cs_generate_variant:Nn \__kernel_backend_matrix:n { x }

(End definition for \__kernel_backend_matrix:n.)

131 </luatex | pdfTeX>

```

### 1.3 dvipdfmx backend

```

132 <*dvipdfmx | xetex>

```

The `dvipdfmx` shares code with the PDF mode one (using the common section to this file) but also with `XYTeX`. The latter is close to identical to `dvipdfmx` and so all of the code here is extracted for both backends, with some `clean up` for `XYTeX` as required. Undocumented but equivalent to pdfTeX’s `literal` keyword. It’s similar to be not the same as the documented `contents` keyword as that adds a `q/Q` pair.

```

\__kernel_backend_literal_pdf:n
\__kernel_backend_literal_pdf:x

```

```

133 \cs_new_protected:Npn \__kernel_backend_literal_pdf:n #1
134 { \__kernel_backend_literal:n { pdf:literal~ #1 } }
135 \cs_generate_variant:Nn \__kernel_backend_literal_pdf:n { x }

(End definition for \__kernel_backend_literal_pdf:n.)

```

```

\__kernel_backend_literal_page:n

```

Whilst the manual says this is like `literal direct` in pdfTeX, it closes the BT block!

```

136 \cs_new_protected:Npn \__kernel_backend_literal_page:n #1
137 { \__kernel_backend_literal:n { pdf:literal~direct~ #1 } }

(End definition for \__kernel_backend_literal_page:n.)

```

```

\__kernel_backend_scope_begin:
\__kernel_backend_scope_end:

```

Scoping is done using the backend-specific specials. We use the versions originally from `xdvipfmx (x:)` as these are well-tested “in the wild”.

```

138 \cs_new_protected:Npn \__kernel_backend_scope_begin:
139 { \__kernel_backend_literal:n { x:gsave } }
140 \cs_new_protected:Npn \__kernel_backend_scope_end:
141 { \__kernel_backend_literal:n { x:grestore } }

```

(End definition for \\_\_kernel\_backend\_scope\_begin: and \\_\_kernel\_backend\_scope\_end:.)

```

142 <@@=sys>

```

```

\c__kernel_sys_dvipdfmx_version_int

```

A short excursion into the `sys` module to set up the backend version information.

```

143 \group_begin:
144 \cs_set:Npn \__sys_tmp:w #1 Version ~ #2 ~ #3 \q_stop {#2}
145 \sys_get_shell:nnTF { extractbb---version }
146 { \char_set_catcode_space:n { ‘\ } }
147 \l_sys_internal_tl
148 {
149     \int_const:Nn \c__kernel_sys_dvipdfmx_version_int
150     {
151         \exp_after:wN \__sys_tmp:w \l_sys_internal_tl
152         \q_stop
153     }
154 }
155 { \int_const:Nn \c__kernel_sys_dvipdfmx_version_int { 0 } }
156 \group_end:

```

(End definition for `\c__kernel_sys_dvipdfmx_version_int`.)

```
157 <@@=)
158 </dvipdfmx | xetex>
```

## 1.4 dvisvgm backend

```
159 <*dvisvgm>
```

```
\_kernel_backend_literal_svg:n
\_kernel_backend_literal_svg:x
```

Unlike the other backends, the requirements for making SVG files mean that we can't conveniently transform all operations to the current point. That makes life a bit more tricky later as that needs to be accounted for. A new line is added after each call to help to keep the output readable for debugging.

```
160 \cs_new_protected:Npn \_kernel_backend_literal_svg:n #1
161 { \_kernel_backend_literal:n { dvisvgm:raw~ #1 { ?nl } } }
162 \cs_generate_variant:Nn \_kernel_backend_literal_svg:n { x }
```

(End definition for `\_kernel_backend_literal_svg:n`.)

```
\g__kernel_backend_scope_int
\l__kernel_backend_scope_int
```

In SVG, we need to track scope nesting as properties attach to scopes; that requires a pair of `int` registers.

```
163 \int_new:N \g__kernel_backend_scope_int
164 \int_new:N \l__kernel_backend_scope_int
```

(End definition for `\g__kernel_backend_scope_int` and `\l__kernel_backend_scope_int`.)

```
\_kernel_backend_scope_begin:
\_kernel_backend_scope_end:
\_kernel_backend_scope_begin:n
\_kernel_backend_scope_begin:x
\_kernel_backend_scope:n
\_kernel_backend_scope:x
```

In SVG, the need to attach concepts to a scope means we need to be sure we will close all of the open scopes. That is easiest done if we only need an outer “wrapper” `begin/end` pair, and within that we apply operations as a simple scoped statements. To keep down the non-productive groups, we also have a `begin` version that does take an argument.

```
165 \cs_new_protected:Npn \_kernel_backend_scope_begin:
166 {
167   \_kernel_backend_literal_svg:n { <g> }
168   \int_set_eq:NN
169     \l__kernel_backend_scope_int
170     \g__kernel_backend_scope_int
171   \group_begin:
172     \int_gset:Nn \g__kernel_backend_scope_int { 1 }
173 }
174 \cs_new_protected:Npn \_kernel_backend_scope_end:
175 {
176   \prg_replicate:nn
177     { \g__kernel_backend_scope_int }
178     { \_kernel_backend_literal_svg:n { </g> } }
179   \group_end:
180   \int_gset_eq:NN
181     \g__kernel_backend_scope_int
182     \l__kernel_backend_scope_int
183 }
184 \cs_new_protected:Npn \_kernel_backend_scope_begin:n #1
185 {
186   \_kernel_backend_literal_svg:n { <g ~ #1 > }
187   \int_set_eq:NN
188     \l__kernel_backend_scope_int
```

```

189     \g__kernel_backend_scope_int
190     \group_begin:
191     \int_gset:Nn \g__kernel_backend_scope_int { 1 }
192   }
193   \cs_generate_variant:Nn \__kernel_backend_scope_begin:n { x }
194   \cs_new_protected:Npn \__kernel_backend_scope:n #1
195   {
196     \__kernel_backend_literal_svg:n { <g ~ #1 > }
197     \int_gincr:N \g__kernel_backend_scope_int
198   }
199   \cs_generate_variant:Nn \__kernel_backend_scope:n { x }

```

(End definition for \\_\_kernel\_backend\_scope\_begin: and others.)

```

200 </dvisvgm>
201 </package>

```

## 2 l3backend-box Implementation

```

202 <*package>
203 <@@=box>

```

### 2.1 dvips backend

```

204 <*dvips>

```

\\_\_box\_backend\_clip:N The **dvips** backend scales all absolute dimensions based on the output resolution selected and any TeX magnification. Thus for any operation involving absolute lengths there is a correction to make. See **normalscale** from **special.pro** for the variables, noting that here everything is saved on the stack rather than as a separate variable. Once all of that is done, the actual clipping is trivial.

```

205   \cs_new_protected:Npn \__box_backend_clip:N #1
206   {
207     \__kernel_backend_scope_begin:
208     \__kernel_backend_align_begin:
209     \__kernel_backend_literal_postscript:n { matrix~currentmatrix }
210     \__kernel_backend_literal_postscript:n
211     { Resolution~72~div~VResolution~72~div~scale }
212     \__kernel_backend_literal_postscript:n { DVImag~dup~scale }
213     \__kernel_backend_literal_postscript:x
214     {
215       0 ~
216       \dim_to_decimal_in_bp:n { \box_dp:N #1 } ~
217       \dim_to_decimal_in_bp:n { \box_wd:N #1 } ~
218       \dim_to_decimal_in_bp:n { -\box_ht:N #1 - \box_dp:N #1 } ~
219       rectclip
220     }
221     \__kernel_backend_literal_postscript:n { setmatrix }
222     \__kernel_backend_align_end:
223     \hbox_overlap_right:n { \box_use:N #1 }
224     \__kernel_backend_scope_end:
225     \skip_horizontal:n { \box_wd:N #1 }
226   }

```

(End definition for \\_\_box\_backend\_clip:N.)

`\__box_backend_rotate:Nn`    Rotating using `dvips` does not require that the box dimensions are altered and has a  
`\__box_backend_rotate_aux:Nn` very convenient built-in operation. Zero rotation must be written as 0 not -0 so there is  
a quick test.

```

227 \cs_new_protected:Npn \__box_backend_rotate:Nn #1#2
228 { \exp_args:NNf \__box_backend_rotate_aux:Nn #1 { \fp_eval:n {#2} } }
229 \cs_new_protected:Npn \__box_backend_rotate_aux:Nn #1#2
230 {
231   \__kernel_backend_scope_begin:
232   \__kernel_backend_align_begin:
233   \__kernel_backend_literal_postscript:x
234   {
235     \fp_compare:nNnTF {#2} = \c_zero_fp
236     { 0 }
237     { \fp_eval:n { round ( -(#2) , 5 ) } } ~
238     rotate
239   }
240   \__kernel_backend_align_end:
241   \box_use:N #1
242   \__kernel_backend_scope_end:
243 }

```

(End definition for `\__box_backend_rotate:Nn` and `\__box_backend_rotate_aux:Nn`.)

`\__box_backend_scale:Nnn`    The `dvips` backend once again has a dedicated operation we can use here.

```

244 \cs_new_protected:Npn \__box_backend_scale:Nnn #1#2#3
245 {
246   \__kernel_backend_scope_begin:
247   \__kernel_backend_align_begin:
248   \__kernel_backend_literal_postscript:x
249   {
250     \fp_eval:n { round ( #2 , 5 ) } ~
251     \fp_eval:n { round ( #3 , 5 ) } ~
252     scale
253   }
254   \__kernel_backend_align_end:
255   \hbox_overlap_right:n { \box_use:N #1 }
256   \__kernel_backend_scope_end:
257 }

```

(End definition for `\__box_backend_scale:Nnn`.)

258 `</dvips>`

## 2.2 LuaTeX and pdfTeX backends

259 `<*luatex | pdftex>`

`\__box_backend_clip:N`    The general method is to save the current location, define a clipping path equivalent to  
the bounding box, then insert the content at the current position and in a zero width box.  
The “real” width is then made up using a horizontal skip before tidying up. There are  
other approaches that can be taken (for example using XForm objects), but the logic here  
shares as much code as possible and uses the same conversions (and so same rounding  
errors) in all cases.

```

260 \cs_new_protected:Npn \__box_backend_clip:N #1

```



```

261 {
262   \_kernel_backend_scope_begin:
263   \_kernel_backend_literal_pdf:x
264   {
265     0~
266     \dim_to_decimal_in_bp:n { -\box_dp:N #1 } ~
267     \dim_to_decimal_in_bp:n { \box_wd:N #1 } ~
268     \dim_to_decimal_in_bp:n { \box_ht:N #1 + \box_dp:N #1 } ~
269     re~W~n
270   }
271   \hbox_overlap_right:n { \box_use:N #1 }
272   \_kernel_backend_scope_end:
273   \skip_horizontal:n { \box_wd:N #1 }
274 }

```

(End definition for \\_box\_backend\_clip:N.)

\\_box\_backend\_rotate:Nn Rotations are set using an affine transformation matrix which therefore requires sine/cosine values not the angle itself. We store the rounded values to avoid rounding twice. There are also a couple of comparisons to ensure that -0 is not written to the output, as this avoids any issues with problematic display programs. Note that numbers are compared to 0 after rounding.

```

275 \cs_new_protected:Npn \_box_backend_rotate:Nn #1#2
276 { \exp_args:Nnf \_box_backend_rotate_aux:Nn #1 { \fp_eval:n {#2} } }
277 \cs_new_protected:Npn \_box_backend_rotate_aux:Nn #1#2
278 {
279   \_kernel_backend_scope_begin:
280   \box_set_wd:Nn #1 { Opt }
281   \fp_set:Nn \l__box_backend_cos_fp { round ( cosd ( #2 ) , 5 ) }
282   \fp_compare:nNnT \l__box_backend_cos_fp = \c_zero_fp
283   { \fp_zero:N \l__box_backend_cos_fp }
284   \fp_set:Nn \l__box_backend_sin_fp { round ( sind ( #2 ) , 5 ) }
285   \_kernel_backend_matrix:x
286   {
287     \fp_use:N \l__box_backend_cos_fp \c_space_tl
288     \fp_compare:nNnTF \l__box_backend_sin_fp = \c_zero_fp
289     { 0~0 }
290     {
291       \fp_use:N \l__box_backend_sin_fp
292       \c_space_tl
293       \fp_eval:n { -\l__box_backend_sin_fp }
294     }
295     \c_space_tl
296     \fp_use:N \l__box_backend_cos_fp
297   }
298   \box_use:N #1
299   \_kernel_backend_scope_end:
300 }
301 \fp_new:N \l__box_backend_cos_fp
302 \fp_new:N \l__box_backend_sin_fp

```

(End definition for \\_box\_backend\_rotate:Nn and others.)

\\_box\_backend\_scale:Nnn The same idea as for rotation but without the complexity of signs and cosines.

```

303 \cs_new_protected:Npn \__box_backend_scale:Nnn #1#2#3
304 {
305   \__kernel_backend_scope_begin:
306   \__kernel_backend_matrix:x
307   {
308     \fp_eval:n { round ( #2 , 5 ) } ~
309     0~0~
310     \fp_eval:n { round ( #3 , 5 ) }
311   }
312   \hbox_overlap_right:n { \box_use:N #1 }
313   \__kernel_backend_scope_end:
314 }

```

(End definition for \\_\_box\_backend\_scale:Nnn.)

```

315 </luatex | pdftex>

```

## 2.3 dvipdfmx/X<sub>Y</sub>TeX backend

```

316 < *dvipdfmx | xetex>

```

\\_\_box\_backend\_clip:N The code here is identical to that for LuaTeX/pdfTeX: unlike rotation and scaling, there is no higher-level support in the backend for clipping.

```

317 \cs_new_protected:Npn \__box_backend_clip:N #1
318 {
319   \__kernel_backend_scope_begin:
320   \__kernel_backend_literal_pdf:x
321   {
322     0~
323     \dim_to_decimal_in_bp:n { -\box_dp:N #1 } ~
324     \dim_to_decimal_in_bp:n { \box_wd:N #1 } ~
325     \dim_to_decimal_in_bp:n { \box_ht:N #1 + \box_dp:N #1 } ~
326     re~W~n
327   }
328   \hbox_overlap_right:n { \box_use:N #1 }
329   \__kernel_backend_scope_end:
330   \skip_horizontal:n { \box_wd:N #1 }
331 }

```

(End definition for \\_\_box\_backend\_clip:N.)

\\_\_box\_backend\_rotate:Nn Rotating in dvipdfmx/X<sub>Y</sub>TeX can be implemented using either PDF or backend-specific code. The former approach however is not “aware” of the content of boxes: this means that any embedded links would not be adjusted by the rotation. As such, the backend-native approach is preferred: the code therefore is similar (though not identical) to the dvips version (notice the rotation angle here is positive). As for dvips, zero rotation is written as 0 not -0.

```

332 \cs_new_protected:Npn \__box_backend_rotate:Nn #1#2
333 { \exp_args:NNf \__box_backend_rotate_aux:Nn #1 { \fp_eval:n {#2} } }
334 \cs_new_protected:Npn \__box_backend_rotate_aux:Nn #1#2
335 {
336   \__kernel_backend_scope_begin:
337   \__kernel_backend_literal:x
338   {
339     x:rotate~

```

```

340     \fp_compare:nNnTF {#2} = \c_zero_fp
341     { 0 }
342     { \fp_eval:n { round ( #2 , 5 ) } } }
343   }
344   \box_use:N #1
345   \__kernel_backend_scope_end:
346 }

```

(End definition for `\__box_backend_rotate:Nn` and `\__box_backend_rotate_aux:Nn`.)

`\__box_backend_scale:Nnn` Much the same idea for scaling: use the higher-level backend operation to allow for box content.

```

347 \cs_new_protected:Npn \__box_backend_scale:Nnn #1#2#3
348 {
349   \__kernel_backend_scope_begin:
350   \__kernel_backend_literal:x
351   {
352     x:scale~
353     \fp_eval:n { round ( #2 , 5 ) } ~
354     \fp_eval:n { round ( #3 , 5 ) }
355   }
356   \hbox_overlap_right:n { \box_use:N #1 }
357   \__kernel_backend_scope_end:
358 }

```

(End definition for `\__box_backend_scale:Nnn`.)

359 `</dvipdfmx | xetex>`

## 2.4 dvisvgm backend

360 `<*dvisvgm>`

`\__box_backend_clip:N`  
`\g__box_clip_path_int`

Clipping in SVG is more involved than with other backends. The first issue is that the clipping path must be defined separately from where it is used, so we need to track how many paths have applied. The naming here uses `l3cp` as the namespace with a number following. Rather than use a rectangular operation, we define the path manually as this allows it to have a depth: easier than the alternative approach of shifting content up and down using scopes to allow for the depth of the `TeX` box and keep the reference point the same!

```

361 \cs_new_protected:Npn \__box_backend_clip:N #1
362 {
363   \int_gincr:N \g__box_clip_path_int
364   \__kernel_backend_literal_svg:x
365   { < clipPath-id = " l3cp \int_use:N \g__box_clip_path_int " > }
366   \__kernel_backend_literal_svg:x
367   {
368     <
369     path ~ d =
370     "
371     M ~ 0 ~
372     \dim_to_decimal:n { -\box_dp:N #1 } ~
373     L ~ \dim_to_decimal:n { \box_wd:N #1 } ~
374     \dim_to_decimal:n { -\box_dp:N #1 } ~
375     L ~ \dim_to_decimal:n { \box_wd:N #1 } ~

```

```

376         \dim_to_decimal:n { \box_ht:N #1 + \box_dp:N #1 } ~
377         L ~ 0 ~
378         \dim_to_decimal:n { \box_ht:N #1 + \box_dp:N #1 } ~
379         Z
380         "
381     />
382 }
383 \__kernel_backend_literal_svg:n
384 { < /clipPath > }

```

In general the SVG set up does not try to transform coordinates to the current point. For clipping we need to do that, so have a transformation here to get us to the right place, and a matching one just before the  $\text{\TeX}$  box is inserted to get things back on track. The clip path needs to come between those two such that if lines up with the current point, as does the  $\text{\TeX}$  box.

```

385 \__kernel_backend_scope_begin:n
386 {
387     transform =
388     "
389         translate ( { ?x } , { ?y } ) ~
390         scale ( 1 , -1 )
391     "
392 }
393 \__kernel_backend_scope:x
394 {
395     clip-path =
396     "url ( \c_hash_str l3cp \int_use:N \g__box_clip_path_int ) "
397 }
398 \__kernel_backend_scope:n
399 {
400     transform =
401     "
402         scale ( -1 , 1 ) ~
403         translate ( { ?x } , { ?y } ) ~
404         scale ( -1 , -1 )
405     "
406 }
407 \box_use:N #1
408 \__kernel_backend_scope_end:
409 }
410 \int_new:N \g__box_clip_path_int

```

(End definition for  $\backslash\_box\_backend\_clip:N$  and  $\backslash g\_box\_clip\_path\_int$ .)

$\backslash\_box\_backend\_rotate:Nn$  Rotation has a dedicated operation which includes a centre-of-rotation optional pair. That can be picked up from the backend syntax, so there is no need to worry about the transformation matrix.

```

411 \cs_new_protected:Npn \__box_backend_rotate:Nn #1#2
412 {
413     \__kernel_backend_scope_begin:x
414     {
415         transform =
416         "
417         rotate

```

```

418      ( \fp_eval:n { round ( -(#2) , 5 ) } , ~ { ?x } , ~ { ?y } )
419      "
420    }
421    \box_use:N #1
422    \__kernel_backend_scope_end:
423  }

```

(End definition for \\_\_box\_backend\_rotate:Nn.)

\\_\_box\_backend\_scale:Nnn In contrast to rotation, we have to account for the current position in this case. That is done using a couple of translations in addition to the scaling (which is therefore done backward with a flip).

```

424 \cs_new_protected:Npn \__box_backend_scale:Nnn #1#2#3
425 {
426   \__kernel_backend_scope_begin:x
427   {
428     transform =
429     "
430       translate ( { ?x } , { ?y } ) ~
431       scale
432       (
433         \fp_eval:n { round ( -#2 , 5 ) } ,
434         \fp_eval:n { round ( -#3 , 5 ) }
435       ) ~
436       translate ( { ?x } , { ?y } ) ~
437       scale ( -1 )
438     "
439   }
440   \hbox_overlap_right:n { \box_use:N #1 }
441   \__kernel_backend_scope_end:
442 }

```

(End definition for \\_\_box\_backend\_scale:Nnn.)

```

443 </dvisvgm>
444 </package>

```

### 3 l3backend-color Implementation

```

445 <*package>
446 <@@=color>

```

Color support is split into parts: collecting data from L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, the color stack, general color, separations, and color for drawings. We have different approaches in each backend, and have some choices to make about dvipdfmx/X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X in particular. Whilst it is in some ways convenient to use the same approach in multiple backends, the fact that dvipdfmx/X<sub>Y</sub>L<sub>A</sub>T<sub>E</sub>X is PDF-based means it (largely) sticks closer to direct PDF output.

#### 3.1 Collecting information from L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

##### 3.1.1 dvips-style

```

447 <*dvisvgm | dvipdfmx | dvips | xetex>

```

`\_color_backend_pickup:N` Allow for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> color. Here, the possible input values are limited: dvips-style colors can mainly be taken as-is with the exception spot ones (here we need a model and a tint).  
`\_color_backend_pickup:w` The x-type expansion is there to cover the case where xcolor is in use.

```

448 \cs_new_protected:Npn \_color_backend_pickup:N #1 { }
449 \cs_if_exist:cT { ver@color.sty }
450 {
451   \cs_set_protected:Npn \_color_backend_pickup:N #1
452   {
453     \exp_args:NV \tl_if_head_is_space:nTF \current@color
454     {
455       \tl_set:Nx #1
456       {
457         { \exp_after:wN \use:n \current@color }
458         { 1 }
459       }
460     }
461     {
462       \exp_last_unbraced:Nx \_color_backend_pickup:w
463       { \current@color } \s__color_stop #1
464     }
465   }
466   \cs_new_protected:Npn \_color_backend_pickup:w #1 ~ #2 \s__color_stop #3
467   { \tl_set:Nn #3 { {#1} {#2} } }
468 }

```

(End definition for `\_color_backend_pickup:N` and `\_color_backend_pickup:w`.)

```

469 </divisvgn | dvipdfmx | dvips | xetex>

```

### 3.1.2 Lua<sub>T</sub><sub>E</sub>X and pdf<sub>T</sub><sub>E</sub>X

```

470 < *luatex | pdftex>

```

`\_color_backend_pickup:N` The current color in driver-dependent format: pick up the package-mode data if available. We end up converting back and forward in this route as we store our color data in dvips format. The `\current@color` needs to be x-expanded before `\_color_backend_pickup:w` breaks it apart, because for instance xcolor sets it to be instructions to generate a color  
`\_color_backend_pickup:w`

```

471 \cs_new_protected:Npn \_color_backend_pickup:N #1 { }
472 \cs_if_exist:cT { ver@color.sty }
473 {
474   \cs_set_protected:Npn \_color_backend_pickup:N #1
475   {
476     \exp_last_unbraced:Nx \_color_backend_pickup:w
477     { \current@color } ~ 0 ~ 0 ~ 0 \s__color_stop #1
478   }
479   \cs_new_protected:Npn \_color_backend_pickup:w
480   #1 ~ #2 ~ #3 ~ #4 ~ #5 ~ #6 \s__color_stop #7
481   {
482     \str_if_eq:nnTF {#2} { g }
483     { \tl_set:Nn #7 { { gray } {#1} } }
484     {
485       \str_if_eq:nnTF {#4} { rg }
486       { \tl_set:Nn #7 { { rgb } { #1 ~ #2 ~ #3 } } }

```

```

487         {
488         \str_if_eq:nnTF {#5} { k }
489         { \tl_set:Nn #7 { { cmyk } { #1 ~ #2 ~ #3 ~ #4 } } }
490         {
491         \str_if_eq:nnTF {#2} { cs }
492         {
493         \tl_set:Nx #7 { { \use:n #1 } { #5 } }
494         }
495         {
496         \tl_set:Nn #7 { { gray } { 0 } }
497         }
498         }
499     }
500 }
501 }
502 }

```

(End definition for `\__color_backend_pickup:N` and `\__color_backend_pickup:w`.)

503 `</luatex | pdftex>`

## 3.2 The color stack

For PDF-based engines, we have a color stack available inside the specials. This is used for concepts beyond color itself: it is needed to manage the graphics state generally. The exact form depends on the engine, and for `dvipdfmx/XYTeX` the backend version.

### 3.2.1 `dvipdfmx/XYTeX`

504 `<*dvipdfmx | xetex>`

`\__kernel_color_stack_init:Nnn` In (x)`dvipdfmx`, the base color stack is not set up, so we have to force that, as well as providing a mechanism more generally.

`\g__color_stack_int`

```

505 \int_compare:nNnF \c__kernel_sys_dvipdfmx_version_int < { 20201111 }
506 {
507     \int_new:N \g__color_stack_int
508     \cs_new_protected:Npn \__kernel_color_stack_init:Nnn #1#2#3
509     {
510         \int_gincr:N \g__color_stack_int
511         \int_const:Nn #1 { \g__color_stack_int }
512         \__kernel_backend_literal:x
513         {
514             pdfcolorstackinit ~
515             \int_use:N \g__color_stack_int \c_space_tl
516             \tl_if_blank:nF {#2} { #2 ~ }
517             (#3)
518         }
519     }
520 }

```

(End definition for `\__kernel_color_stack_init:Nnn` and `\g__color_stack_int`.)

Simple enough but needs a version check.

```

521 \int_compare:nNnF \c__kernel_sys_dvipdfmx_version_int < { 20201111 }
522 {

```

```

523 \cs_new_protected:Npn \__kernel_color_stack_push:nn #1#2
524 {
525   \__kernel_backend_literal:x
526   {
527     pdfcolorstack ~
528     \int_eval:n {#1} ~
529     push ~ (#2)
530   }
531 }
532 \cs_new_protected:Npn \__kernel_color_stack_pop:n #1
533 {
534   \__kernel_backend_literal:x
535   {
536     pdfcolorstack ~
537     \int_eval:n {#1} ~
538     pop
539   }
540 }
541 }

```

(End definition for \\_\_kernel\_color\_stack\_push:nn and \\_\_kernel\_color\_stack\_pop:n.)

```
542 </dvipdfmx | xetex>
```

### 3.2.2 LuaTeX and pdfTeX

```
543 < *luatex | pdftex >
```

\\_\_kernel\_color\_stack\_init:Nnn

```

544 \cs_new_protected:Npn \__kernel_color_stack_init:Nnn #1#2#3
545 {
546   \int_const:Nn #1
547   {
548     <*luatex>
549     \tex_pdffeedback:D colorstackinit ~
550     </luatex>
551     <*pdftex>
552     \tex_pdfcolorstackinit:D
553     </pdftex>
554     \tl_if_blank:nF {#2} { #2 ~ }
555     {#3}
556   }
557 }

```

(End definition for \\_\_kernel\_color\_stack\_init:Nnn.)

\\_\_kernel\_color\_stack\_push:nn

\\_\_kernel\_color\_stack\_pop:n

```

558 \cs_new_protected:Npn \__kernel_color_stack_push:nn #1#2
559 {
560   <*luatex>
561   \tex_pdfextension:D colorstack ~
562   </luatex>
563   <*pdftex>
564   \tex_pdfcolorstack:D
565   </pdftex>

```



```

566     \int_eval:n {#1} ~ push ~ {#2}
567   }
568   \cs_new_protected:Npn \__kernel_color_stack_pop:n #1
569   {
570     \*luatex
571     \tex_pdfextension:D colorstack ~
572   \*pdfTeX
573   \tex_pdfcolorstack:D
574   \*pdfTeX
575   \int_eval:n {#1} ~ pop \scan_stop:
576 }
577

```

(End definition for \\_\_kernel\_color\_stack\_push:nn and \\_\_kernel\_color\_stack\_pop:n.)

```

578 \</luatex | pdfTeX>

```

### 3.3 General color

#### 3.3.1 dvips-style

```

579 \*dvips | dvisvgm>

```

Push the data to the stack. In the case of dvips also saves the drawing color in raw PostScript.

```

\__color_backend_select_cmyk:n
\__color_backend_select_gray:n
\__color_backend_select_rgb:n
\__color_backend_select:n
\__color_backend_reset:
color.sc
color.fc
580 \cs_new_protected:Npn \__color_backend_select_cmyk:n #1
581 { \__color_backend_select:n { cmyk ~ #1 } }
582 \cs_new_protected:Npn \__color_backend_select_gray:n #1
583 { \__color_backend_select:n { gray ~ #1 } }
584 \cs_new_protected:Npn \__color_backend_select_rgb:n #1
585 { \__color_backend_select:n { rgb ~ #1 } }
586 \cs_new_protected:Npn \__color_backend_select:n #1
587 {
588   \__kernel_backend_literal:n { color~push~ #1 }
589 \*dvips
590   \__kernel_backend_postscript:n { /color.sc~ { ~ } ~def }
591   \__kernel_backend_postscript:n { /color.fc~ { ~ } ~def }
592 \</dvips>
593   \group_insert_after:N \__color_backend_reset:
594 }
595 \cs_new_protected:Npn \__color_backend_reset:
596 { \__kernel_backend_literal:n { color~pop } }

```

(End definition for \\_\_color\_backend\_select\_cmyk:n and others. These functions are documented on page ??.)

```

597 \</dvips | dvisvgm>

```

#### 3.3.2 LuaTeX and pdfTeX

```

598 \*dviPDFmx | luatex | pdfTeX | xetex>

```

\l\_\_kernel\_color\_stack\_int pdfTeX, LuaTeX and recent (x)dviPDFmx have multiple stacks available, and to track which one is in use a variable is required.

```

599 \int_new:N \l__kernel_color_stack_int

```

(End definition for \l\_\_kernel\_color\_stack\_int.)

```

\__color_backend_select_cmyk:n
\__color_backend_select_gray:n
\__color_backend_select_rgb:n
\__color_backend_reset:

```

Simply dump the data, but allowing for LuaTeX.

```

600 \cs_new_protected:Npn \__color_backend_select_cmyk:n #1
601 { \__color_backend_select:n { #1 ~ k ~ #1 ~ K } }
602 \cs_new_protected:Npn \__color_backend_select_gray:n #1
603 { \__color_backend_select:n { #1 ~ g ~ #1 ~ G } }
604 \cs_new_protected:Npn \__color_backend_select_rgb:n #1
605 { \__color_backend_select:n { #1 ~ rg ~ #1 ~ RG } }
606 \cs_new_protected:Npn \__color_backend_select:n #1
607 {
608   \__kernel_color_stack_push:nn \l__kernel_color_stack_int {#1}
609   \group_insert_after:N \__color_backend_reset:
610 }
611 \cs_new_protected:Npn \__color_backend_reset:
612 { \__kernel_color_stack_pop:n \l__kernel_color_stack_int }

```

(End definition for \\_\_color\_backend\_select\_cmyk:n and others.)

```

613 </dvipdfmx | luatex | pdftex | xetex>

```

### 3.3.3 dvipdfmx/X<sub>Y</sub>TeX

```

614 <*dvipdfmx | xetex>

```

These backends have the most possible approaches: it recognises both dvips-based color specials and it’s own format, plus one can include PDF statements directly. Recent releases also have a color stack approach similar to pdfTeX. Of the stack methods, the dedicated the most versatile is the latter as it can cover all of the use cases we have. Thus it is used in preference to the dvips-style interface or the “native” color specials (which have only one stack).

```

\__color_backend_select_cmyk:n
\__color_backend_select_gray:n
\__color_backend_select_rgb:n
\__color_backend_reset:

```

Push the data to the stack.

```

615 \int_compare:nNnT \c__kernel_sys_dvipdfmx_version_int < { 20201111 }
616 {
617   \cs_gset_protected:Npn \__color_backend_select_cmyk:n #1
618   {
619     \__kernel_backend_literal:n { pdf: bc ~ [#1] }
620     \group_insert_after:N \__color_backend_reset:
621   }
622   \cs_gset_eq:NN \__color_backend_select_gray:n \__color_backend_select_cmyk:n
623   \cs_gset_eq:NN \__color_backend_select_rgb:n \__color_backend_select_cmyk:n
624   \cs_gset_protected:Npn \__color_backend_reset:
625   { \__kernel_backend_literal:n { pdf: ec } }
626 }

```

(End definition for \\_\_color\_backend\_select\_cmyk:n and others.)

```

627 </dvipdfmx | xetex>

```

## 3.4 Separations

Here, life gets interesting and we need essentially one approach per backend.

```

628 <*dvips>

```

```

\__color_backend_select_separation:nn
\__color_backend_select_devicen:nn

```

```

629 \cs_new_protected:Npn \__color_backend_select_separation:nn #1#2
630 { \__color_backend_select:n { separation ~ #1 ~ #2 } }
631 \cs_new_eq:NN \__color_backend_select_devicen:nn \__color_backend_select_separation:nn

```

(End definition for `__color_backend_select_separation:nn` and `__color_backend_select_device:nn`.)

```

__color_backend_separation_init:nnnnn
__color_backend_separation_init:nxxnn
__color_backend_separation_init_aux:nnnnn
__color_backend_separation_init_DeviceCMYK:nnn
__color_backend_separation_init_DeviceGray:nnn
__color_backend_separation_init_DeviceRGB:nnn
__color_backend_separation_init_Device:Nn
__color_backend_separation_init:nnn
__color_backend_separation_init_count:n
__color_backend_separation_init_count:w
__color_backend_separation_init:nnnnn
__color_backend_separation_init:w
__color_backend_separation_init:n
__color_backend_separation_init:nnw
__color_backend_separation_init_CIELAB:nnn

```

Initialising here means creating a small header set up plus massaging some data. This comes about as we have to deal with PDF-focussed data, which makes most sense “higher-up”. The approach is based on ideas from <https://tex.stackexchange.com/q/560093> plus using the PostScript manual for other aspects.

```

632 \cs_new_protected:Npx __color_backend_separation_init:nnnnn #1#2#3#4#5
633 {
634   \bool_if:NT \g__kernel_backend_header_bool
635   {
636     \cs_if_exist:NTF \AtBeginDvi
637     { \exp_not:N \AtBeginDvi }
638     { \use:n }
639     {
640       \exp_not:N __color_backend_separation_init_aux:nnnnn
641       {#1} {#2} {#3} {#4} {#5}
642     }
643   }
644 }
645 \cs_generate_variant:Nn __color_backend_separation_init:nnnnn { nxx }
646 \cs_new_protected:Npn __color_backend_separation_init_aux:nnnnn #1#2#3#4#5
647 {
648   __kernel_backend_literal:e
649   {
650     !
651     TeXDict ~ begin ~
652     /color \int_use:N \g__color_model_int
653     {
654       [ ~
655       /Separation ~ ( \str_convert_pdfname:n {#1} ) ~
656       [ ~ #2 ~ ] ~
657       {
658         \cs_if_exist_use:cF { __color_backend_separation_init_ #2 :nnn }
659         { __color_backend_separation_init:nnn }
660         {#3} {#4} {#5}
661       }
662     ] ~ setcolorspace
663     } ~ def ~
664   end
665 }
666 }
667 \cs_new:cpn { __color_backend_separation_init_ /DeviceCMYK :nnn } #1#2#3
668 { __color_backend_separation_init_Device:Nn 4 {#3} }
669 \cs_new:cpn { __color_backend_separation_init_ /DeviceGray :nnn } #1#2#3
670 { __color_backend_separation_init_Device:Nn 1 {#3} }
671 \cs_new:cpn { __color_backend_separation_init_ /DeviceRGB :nnn } #1#2#3
672 { __color_backend_separation_init_Device:Nn 2 {#3} }
673 \cs_new:Npn __color_backend_separation_init_Device:Nn #1#2
674 {
675   #2 ~
676   \prg_replicate:nn {#1}
677   { #1 ~ index ~ mul ~ #1 ~ 1 ~ roll ~ }
678   \int_eval:n { #1 + 1 } ~ -1 ~ roll ~ pop
679 }

```

For the generic case, we cannot use `/FunctionType 2` unfortunately, so we have to code that idea up in PostScript. Here, we will therefore assume that a range is *always* given. First, we count values in each argument: at the backend level, we can assume there are always well-behaved with spaces present.

```

680 \cs_new:Npn \__color_backend_separation_init:nnn #1#2#3
681 {
682   \exp_args:Ne \__color_backend_separation_init:nnnn
683   { \__color_backend_separation_init_count:n {#2} }
684   {#1} {#2} {#3}
685 }
686 \cs_new:Npn \__color_backend_separation_init_count:n #1
687 { \int_eval:n { 0 \__color_backend_separation_init_count:w #1 ~ \s__color_stop } }
688 \cs_new:Npn \__color_backend_separation_init_count:w #1 ~ #2 \s__color_stop
689 {
690   +1
691   \tl_if_blank:nF {#2}
692   { \__color_backend_separation_init_count:w #2 \s__color_stop }
693 }

```

Now we implement the algorithm. In the terms in the PostScript manual, we have  $\mathbf{N} = 1$  and  $\mathbf{Domain} = [0 \ 1]$ , with  $\mathbf{Range}$  as  $\#2$ ,  $\mathbf{C0}$  as  $\#3$  and  $\mathbf{C1}$  as  $\#4$ , with the number of output components in  $\#1$ . So all we have to do is implement  $y_i = \mathbf{C0}_i + x(\mathbf{C1}_i - \mathbf{C0}_i)$  with lots of stack manipulation, then check the ranges. That's done by adding everything to the stack first, then using the fact we know all of the offsets. As manipulating the stack is tricky, we start by re-formatting the  $\mathbf{C0}$  and  $\mathbf{C1}$  arrays to be interleaved, and add a 0 to each pair: this is used to keep the stack of constant length while we are doing the first pass of mathematics. We then working through that list, calculating from the last to the first value before tidying up by removing all of the input values. We do that by first copying all of the final  $y$  values to the end of the stack, then rolling everything so we can pop the now-unneeded material.

```

694 \cs_new:Npn \__color_backend_separation_init:nnnn #1#2#3#4
695 {
696   \__color_backend_separation_init:w #3 ~ \s__color_stop #4 ~ \s__color_stop
697   \prg_replicate:nn {#1}
698   {
699     pop ~ 1 ~ index ~ neg ~ 1 ~ index ~ add ~
700     \int_eval:n { 3 * #1 } ~ index ~ mul ~
701     2 ~ index ~ add ~
702     \int_eval:n { 3 * #1 } ~ #1 ~ roll ~
703   }
704   \int_step_function:nnnN {#1} { -1 } { 1 }
705   \__color_backend_separation_init:n
706   \int_eval:n { 4 * #1 + 1 } ~ #1 ~ roll ~
707   \prg_replicate:nn { 3 * #1 + 1 } { pop ~ }
708   \tl_if_blank:nF {#2}
709   { \__color_backend_separation_init:nw {#1} #2 ~ \s__color_stop }
710 }
711 \cs_new:Npn \__color_backend_separation_init:w
712 #1 ~ #2 \s__color_stop #3 ~ #4 \s__color_stop
713 {
714   #1 ~ #3 ~ 0 ~
715   \tl_if_blank:nF {#2}
716   { \__color_backend_separation_init:w #2 \s__color_stop #4 \s__color_stop }

```

```

717 }
718 \cs_new:Npn \__color_backend_separation_init:n #1
719 { \int_eval:n { #1 * 2 } ~ index ~ }

```

Finally, we deal with the range limit if required. This is handled by splitting the range into pairs. It's then just a question of doing the comparisons, this time dropping everything except the desired result.

```

720 \cs_new:Npn \__color_backend_separation_init:nw #1#2 ~ #3 ~ #4 \s_color_stop
721 {
722   #2 ~ #3 ~
723   2 ~ index ~ 2 ~ index ~ lt ~
724   { ~ pop ~ exch ~ pop ~ } ~
725   { ~
726     2 ~ index ~ 1 ~ index ~ gt ~
727     { ~ exch ~ pop ~ exch ~ pop ~ } ~
728     { ~ pop ~ pop ~ } ~
729     ifelse ~
730   }
731   ifelse ~
732   #1 ~ 1 ~ roll ~
733   \tl_if_blank:nF {#4}
734   { \__color_backend_separation_init:nw {#1} #4 \s_color_stop }
735 }

```

CIELAB support uses the detail from the PostScript reference, page 227; other than that block of PostScript, this is the same as for PDF-based routes.

```

736 \cs_new_protected:Npn \__color_backend_separation_init_CIELAB:nnn #1#2#3
737 {
738   \__color_backend_separation_init:nxxxnn
739   {#2}
740   {
741     /CIEBasedABC ~
742     << ~
743     /RangeABC ~ [ ~ \c__color_model_range_CIELAB_tl \c_space_tl ] ~
744     /DecodeABC ~
745     [ ~
746       { ~ 16 ~ add ~ 116 ~ div ~ } ~ bind ~
747       { ~ 500 ~ div ~ } ~ bind ~
748       { ~ 200 ~ div ~ } ~ bind ~
749     ] ~
750     /MatrixABC ~ [ ~ 1 ~ 1 ~ 1 ~ 1 ~ 0 ~ 0 ~ 0 ~ 0 ~ -1 ~ ] ~
751     /DecodeLMN ~
752     [ ~
753     { ~
754       dup ~ 6 ~ 29 ~ div ~ ge ~
755       { ~ dup ~ dup ~ mul ~ mul ~ ~ } ~
756       { ~ 4 ~ 29 ~ div ~ sub ~ 108 ~ 841 ~ div ~ mul ~ } ~
757       ifelse ~
758       0.9505 ~ mul ~
759     } ~ bind ~
760     { ~
761       dup ~ 6 ~ 29 ~ div ~ ge ~
762       { ~ dup ~ dup ~ mul ~ mul ~ ~ } ~
763       { ~ 4 ~ 29 ~ div ~ sub ~ 108 ~ 841 ~ div ~ mul ~ } ~
764       ifelse ~

```

```

765         } ~ bind ~
766         { ~
767             dup ~ 6 ~ 29 ~ div ~ ge ~
768             { ~ dup ~ dup ~ mul ~ mul ~ } ~
769             { ~ 4 ~ 29 ~ div ~ sub ~ 108 ~ 841 ~ div ~ mul ~ } ~
770             ifelse ~
771             1.0890 ~ mul ~
772         } ~ bind
773     ] ~
774     /WhitePoint ~
775     [ ~ \tl_use:c { c__color_model_whitepoint_CIELAB_ #1 _tl } ~ ] ~
776     >>
777 }
778 { \c__color_model_range_CIELAB_tl }
779 { 100 ~ 0 ~ 0 }
780 {#3}
781 }

```

(End definition for \\_color\_backend\_separation\_init:nnnnn and others.)

\\_color\_backend\_devicen\_init:nnn Trivial as almost all of the work occurs in the shared code.

```

782 \cs_new_protected:Npn \_color_backend_devicen_init:nnn #1#2#3
783 {
784     \_kernel_backend_literal:e
785     {
786         !
787         TeXDict ~ begin ~
788         /color \int_use:N \g__color_model_int
789         {
790             [ ~
791                 /DeviceN ~
792                 [ ~ #1 ~ ] ~
793                 #2 ~
794                 { ~ #3 ~ } ~
795             ] ~ setcolorspace
796         } ~ def ~
797     end
798 }
799 }

```

(End definition for \\_color\_backend\_devicen\_init:nnn.)

```

800 </dvips>
801 <*dvisvgm>

```

\\_color\_backend\_select\_separation:nn No support at present.

```

\__color_backend_select_devicen:nn
802 \cs_new_protected:Npn \_color_backend_select_separation:nn #1#2 { }
803 \cs_new_protected:Npn \_color_backend_select_devicen:nn #1#2 { }

```

(End definition for \\_color\_backend\_select\_separation:nn and \\_color\_backend\_select\_devicen:nn.)

\\_color\_backend\_separation\_init:nnnnn No support at present.

```

\__color_backend_separation_init_CIELAB:nnn
804 \cs_new_protected:Npn \_color_backend_separation_init:nnnnn #1#2#3#4#5 { }
805 \cs_new_protected:Npn \_color_backend_separation_init_CIELAB:nnnnnn #1#2#3 { }

```

(End definition for `\_color_backend_separation_init:nnnnn` and `\_color_backend_separation_init_CIELAB:nnn`.)

806 `</dvisvgm>`

807 `<*dvipdfmx | luatex | pdftex | xetex>`

`\_color_backend_select_separation:nn`  
`\_color_backend_select_devicen:nn`  
`\_color_backend_select:n`

Although (x)dvipdfmx has a built-in approach to color spaces, that can't be used with the generic color stacks. So we take an approach in which we share the same code as for pdfTeX.

808 `\cs_new_protected:Npn \_color_backend_select_separation:nn #1#2`  
809 `{ \_color_backend_select:n { /#1 ~ cs ~ /#1 ~ CS ~ #2 ~ scn ~ #2 ~ SCN } }`  
810 `\cs_new_eq:NN \_color_backend_select_devicen:nn \_color_backend_select_separation:nn`

(End definition for `\_color_backend_select_separation:nn`, `\_color_backend_select_devicen:nn`, and `\_color_backend_select:n`.)

`\_color_backend_separation_init:nnnnn`  
`\_color_backend_separation_init:n`  
`\_color_backend_separation_init_CIELAB:nnn`

Initialising the PDF structures needs two parts: creating an object containing the “real” name of the Separation, then adding a reference to that to each page. We use a separate object for the tint transformation following the model in the PDF reference.

811 `\cs_new_protected:Npn \_color_backend_separation_init:nnnnn #1#2#3#4#5`  
812 `{`  
813 `\pdf_object_now:nx { dict }`  
814 `{`  
815 `/FunctionType ~ 2`  
816 `/Domain ~ [0 ~ 1]`  
817 `\tl_if_blank:nF {#3} { /Range ~ [#3] }`  
818 `/C0 ~ [#4] ~`  
819 `/C1 ~ [#5] /N ~ 1`  
820 `}`  
821 `\_color_backend_separation_init:n`  
822 `{`  
823 `/Separation ~`  
824 `/ \str_convert_pdfname:n {#1} ~ #2 ~`  
825 `\pdf_object_last:`  
826 `}`  
827 `\use:x`  
828 `{`  
829 `\pdfmanagement_add:nnn`  
830 `{ Page / Resources / ColorSpace }`  
831 `{ color \int_use:N \g_color_model_int }`  
832 `{ \pdf_object_last: }`  
833 `}`  
834 `}`  
835 `\cs_if_exist:NF \pdf_object_now:nn`  
836 `{ \cs_gset_protected:Npn \_color_backend_separation_init:nnnnn #1#2#3#4#5 { } }`  
837 `\cs_new_protected:Npn \_color_backend_separation_init:n #1`  
838 `{`  
839 `\pdf_object_now:nx { array } {#1}`  
840 `}`

For CIELAB colors, we need one object per document for the illuminant, plus initialisation of the color space referencing that object.

841 `\cs_new_protected:Npn \_color_backend_separation_init_CIELAB:nnn #1#2#3`  
842 `{`

```

843 \pdf_object_if_exist:nF { __color_illuminant_CIELAB_ #1 }
844 {
845   \pdf_object_new:nn { __color_illuminant_CIELAB_ #1 } { array }
846   \pdf_object_write:nx { __color_illuminant_CIELAB_ #1 }
847   {
848     /Lab ~
849     <<
850     /WhitePoint ~
851     [ \tl_use:c { c__color_model_whitepoint_CIELAB_ #1 _tl } ]
852     /Range ~ [ \c__color_model_range_CIELAB_tl ]
853     >>
854   }
855 }
856 \__color_backend_separation_init:nnnnn
857 {#2}
858 { \pdf_object_ref:n { __color_illuminant_CIELAB_ #1 } }
859 { \c__color_model_range_CIELAB_tl }
860 { 100 ~ 0 ~ 0 }
861 {#3}
862 }
863 \cs_if_exist:NF \pdf_object_now:nn
864 {
865   \cs_gset_protected:Npn \__color_backend_separation_init_CIELAB:nnn #1#2#3
866   { }
867 }

```

(End definition for \\_\_color\_backend\_separation\_init:nnnnn, \\_\_color\_backend\_separation\_init:n, and \\_\_color\_backend\_separation\_init\_CIELAB:nnn.)

```

\__color_backend_devicen_init:nnn
\__color_backend_devicen_init:w
\__color_backend_devicen_init:n

```

Similar to the Separations case, but with an arbitrary function for the alternative space work.

```

868 \cs_new_protected:Npn \__color_backend_devicen_init:nnn #1#2#3
869 {
870   \pdf_object_now:nx { stream }
871   {
872     {
873       /FunctionType ~ 4 ~
874       /Domain ~
875       [ ~
876         \prg_replicate:nn
877         { 0 \__color_backend_devicen_init:w #1 ~ \s__color_stop }
878         { 0 ~ 1 ~ } ~
879       ] ~
880       /Range ~
881       [ ~
882         \str_case:nn {#2}
883         {
884           { /DeviceCMYK } { 0 ~ 1 ~ 0 ~ 1 ~ 0 ~ 1 ~ 0 ~ 1 }
885           { /DeviceGray } { 0 ~ 1 }
886           { /DeviceRGB } { 0 ~ 1 ~ 0 ~ 1 ~ 0 ~ 1 }
887         } ~
888       ]
889     }
890     {#3}

```



```

891     }
892     \__color_backend_separation_init:n
893     {
894         /DeviceN ~
895         [ ~ #1 ~ ] ~
896         #2 ~
897         \pdf_object_last:
898     }
899     \use:x
900     {
901         \pdfmanagement_add:nnn
902         { Page / Resources / ColorSpace }
903         { color \int_use:N \g__color_model_int }
904         { \pdf_object_last: }
905     }
906 }
907 \cs_if_exist:NF \pdf_object_now:nn
908 { \cs_gset_protected:Npn \__color_backend_devicen_init:nnn #1#2#3 { } }
909 \cs_new:Npn \__color_backend_devicen_init:w #1 ~ #2 \s__color_stop
910 {
911     + 1
912     \tl_if_blank:NF {#2}
913     { \__color_backend_devicen_init:w #2 \s__color_stop }
914 }
915 \cs_new_eq:NN \__color_backend_devicen_init:n \__color_backend_separation_init:n

```

(End definition for \\_\_color\_backend\_devicen\_init:nnn, \\_\_color\_backend\_devicen\_init:w, and \\_\_color\_backend\_devicen\_init:n.)

```

916 </dvipdfmx | luatex | pdftex | xetex>

```

### 3.5 Fill and stroke color

Here, dvipdfmx/X<sub>q</sub>T<sub>E</sub>X follows LuaT<sub>E</sub>X and pdfT<sub>E</sub>X, while for dvips we have to manage fill and stroke color ourselves. We also handle dvisvgm independently, as there we can create SVG directly.

```

917 < *dvipdfmx | luatex | pdftex | xetex>

```

Drawing (fill/stroke) color is handled in dvipdfmx/X<sub>q</sub>T<sub>E</sub>X in the same way as LuaT<sub>E</sub>X/pdfT<sub>E</sub>X. We use the same approach as earlier, except the color stack is not involved so the generic direct PDF operation is used. There is no worry about the nature of strokes: everything is handled automatically.

```

\__color_backend_fill_cmyk:n
\__color_backend_fill_gray:n
\__color_backend_fill_rgb:n
\__color_backend_stroke_cmyk:n
\__color_backend_stroke_gray:n
\__color_backend_stroke_rgb:n
918 \cs_new_protected:Npn \__color_backend_fill_cmyk:n #1
919 { \__kernel_backend_literal_pdf:n { #1 ~ k } }
920 \cs_new_protected:Npn \__color_backend_fill_gray:n #1
921 { \__kernel_backend_literal_pdf:n { #1 ~ g } }
922 \cs_new_protected:Npn \__color_backend_fill_rgb:n #1
923 { \__kernel_backend_literal_pdf:n { #1 ~ rg } }
924 \cs_new_protected:Npn \__color_backend_stroke_cmyk:n #1
925 { \__kernel_backend_literal_pdf:n { #1 ~ K } }
926 \cs_new_protected:Npn \__color_backend_stroke_gray:n #1
927 { \__kernel_backend_literal_pdf:n { #1 ~ G } }
928 \cs_new_protected:Npn \__color_backend_stroke_rgb:n #1
929 { \__kernel_backend_literal_pdf:n { #1 ~ RG } }

```

(End definition for `\_color_backend_fill_cmyk:n` and others.)

```

\_color_backend_fill_separation:nn
\_color_backend_stroke_separation:nn
  \_color_backend_fill_devicen:nn
  \_color_backend_stroke_devicen:nn
930 \cs_new_protected:Npn \_color_backend_fill_separation:nn #1#2
931   { \_kernel_backend_literal_pdf:n { /#1 ~ cs ~ #2 ~ scn } }
932 \cs_new_protected:Npn \_color_backend_stroke_separation:nn #1#2
933   { \_kernel_backend_literal_pdf:n { /#1 ~ CS ~ #2 ~ SCN } }
934 \cs_new_eq:NN \_color_backend_fill_devicen:nn \_color_backend_fill_separation:nn
935 \cs_new_eq:NN \_color_backend_stroke_devicen:nn \_color_backend_stroke_separation:nn

```

(End definition for `\_color_backend_fill_separation:nn` and others.)

```

936 </dvipdfmx | luatex | pdftex | xetex>
937 <*dvips>

```

All questions of saving the non-stacked data.

```

\_color_backend_fill_cmyk:n
\_color_backend_fill_gray:n
\_color_backend_fill_rgb:n
  \_color_backend_stroke_cmyk:n
  \_color_backend_stroke_gray:n
  \_color_backend_stroke_rgb:n
938 \cs_new_protected:Npn \_color_backend_fill_cmyk:n #1
939   { \_kernel_backend_postscript:n { /color.fc { #1 ~ setcmykcolor } def } }
940 \cs_new_protected:Npn \_color_backend_fill_gray:n #1
941   { \_kernel_backend_postscript:n { /color.fc { #1 ~ setgray } def } }
942 \cs_new_protected:Npn \_color_backend_fill_rgb:n #1
943   { \_kernel_backend_postscript:n { /color.fc { #1 ~ setrgbcolor } def } }
944 \cs_new_protected:Npn \_color_backend_stroke_cmyk:n #1
945   { \_kernel_backend_postscript:n { /color.sc { #1 ~ setcmykcolor } def } }
946 \cs_new_protected:Npn \_color_backend_stroke_gray:n #1
947   { \_kernel_backend_postscript:n { /color.sc { #1 ~ setgray } def } }
948 \cs_new_protected:Npn \_color_backend_stroke_rgb:n #1
949   { \_kernel_backend_postscript:n { /color.sc { #1 ~ setrgbcolor } def } }

```

(End definition for `\_color_backend_fill_cmyk:n` and others.)

```

\_color_backend_fill_separation:nn
\_color_backend_stroke_separation:nn
  \_color_backend_fill_devicen:nn
  \_color_backend_stroke_devicen:nn
950 \cs_new_protected:Npn \_color_backend_fill_separation:nn #1#2
951   { \_kernel_backend_postscript:n { /color.fc { #1 } def } }
952 \cs_new_protected:Npn \_color_backend_stroke_separation:nn #1#2
953   { \_kernel_backend_postscript:n { /color.sc { #1 } def } }
954 \cs_new_eq:NN \_color_backend_fill_devicen:nn \_color_backend_fill_separation:nn
955 \cs_new_eq:NN \_color_backend_stroke_devicen:nn \_color_backend_stroke_separation:nn

```

(End definition for `\_color_backend_fill_separation:nn` and others.)

```

956 </dvips>
957 <*dvisvgm>

```

For drawings in SVG, we use scopes for all colors. That requires using RGB values, which luckily are easy to convert here (cmyk to RGB is a fixed function).

```

\_color_backend_fill_cmyk:n
  \_color_backend_stroke_cmyk:n
  \_color_backend_cmyk:nw
\_color_backend_fill_gray:n
  \_color_backend_stroke_gray:n
  \_color_backend_gray:nn
\_color_backend_gray_aux:n
\_color_backend_fill_rgb:n
  \_color_backend_stroke_rgb:n
  \_color_backend_rgb:nw
  \_color_backend:nnnn
958 \cs_new_protected:Npn \_color_backend_fill_cmyk:n #1
959   { \_color_backend_cmyk:nw { fill } #1 \s_color_stop }
960 \cs_new_protected:Npn \_color_backend_stroke_cmyk:n #1
961   { \_color_backend_cmyk:nw { stroke } #1 \s_color_stop }
962 \cs_new_protected:Npn \_color_backend_cmyk:nw
963   #1#2 ~ #3 ~ #4 ~ #5 \s_color_stop
964   {
965     \use:x
966     {

```

```

967     \_color_backend:nnnn
968     {#1}
969     { \fp_eval:n { -100 * ( 1 - min ( 1 , #2 + #5 ) ) } }
970     { \fp_eval:n { -100 * ( 1 - min ( 1 , #3 + #5 ) ) } }
971     { \fp_eval:n { -100 * ( 1 - min ( 1 , #4 + #5 ) ) } }
972   }
973 }
974 \cs_new_protected:Npn \_color_backend_fill_gray:n #1
975 { \_color_backend_grab:nn { fill } {#1} }
976 \cs_new_protected:Npn \_color_backend_stroke_gray:n #1
977 { \_color_backend_grab:nn { stroke } {#1} }
978 \cs_new_protected:Npn \_color_backend_gray:nn #1#2
979 {
980   \use:x
981   {
982     \_color_backend_gray_aux:nn
983     {#1}
984     { \fp_eval:n { 100 * (#2) } }
985   }
986 }
987 \cs_new_protected:Npn \_color_backend_gray_aux:nn #1#2
988 { \_color_backend:nnnn {#1} {#2} {#2} {#2} }
989 \cs_new_protected:Npn \_color_backend_fill_rgb:n #1
990 { \_color_backend_rgb:nw { fill } #1 \s_color_stop }
991 \cs_new_protected:Npn \_color_backend_stroke_rgb:n #1
992 { \_color_backend_rgb:nw { stroke } #1 \s_color_stop }
993 \cs_new_protected:Npn \_color_backend_rgb:nw
994 #1#2 ~ #3 ~ #4\s_color_stop
995 {
996   \use:x
997   {
998     \_color_backend:nnnn
999     { fill }
1000     { \fp_eval:n { 100 * (#2) } }
1001     { \fp_eval:n { 100 * (#3) } }
1002     { \fp_eval:n { 100 * (#4) } }
1003   }
1004 }
1005 \cs_new_protected:Npx \_color_backend:nnnn #1#2#3#4
1006 {
1007   \_kernel_backend_scope:n
1008   {
1009     #1 =
1010     "
1011     rgb
1012     (
1013       #2 \c_percent_str ,
1014       #3 \c_percent_str ,
1015       #4 \c_percent_str
1016     )
1017     "
1018   }
1019 }

```

(End definition for \\_color\_backend\_fill\_cmyk:n and others.)

```

\__color_backend_fill_separation:nn
\__color_backend_stroke_separation:nn
\__color_backend_fill_devicen:nn
\__color_backend_stroke_devicen:nn

```

At present, these are no-ops.

```

1020 \cs_new_protected:Npn \__color_backend_fill_separation:nn #1#2 { }
1021 \cs_new_protected:Npn \__color_backend_stroke_separation:nn #1#2 { }
1022 \cs_new_eq:NN \__color_backend_fill_devicen:nn \__color_backend_fill_separation:nn
1023 \cs_new_eq:NN \__color_backend_stroke_devicen:nn \__color_backend_stroke_separation:nn

```

(End definition for \\_\_color\_backend\_fill\_separation:nn and others.)

```

1024 </dvisvgm>
1025 </package>

```

## 4 l3backend-draw Implementation

```

1026 <*package>
1027 <@@=draw>

```

### 4.1 dvips backend

```

1028 <*dvips>

```

The same as literal PostScript: same arguments about positioning apply her.

```

1029 \cs_new_eq:NN \__draw_backend_literal:n \__kernel_backend_literal_postscript:n
1030 \cs_generate_variant:Nn \__draw_backend_literal:n { x }

```

(End definition for \\_\_draw\_backend\_literal:n.)

The `ps::[begin]` special here deals with positioning but allows us to continue on to a matching `ps::[end]`: contrast with `ps:`, which positions but where we can't split material between separate calls. The `@beginspecial/@endspecial` pair are from `special.pro` and correct the scale and *y*-axis direction. In contrast to `pgf`, we don't save the current point: discussion with Tom Rokici suggested a better way to handle the necessary translations (see `\__draw_backend_box_use:Nnnnn`). (Note that `@beginspecial/@endspecial` forms a backend scope.) The `[begin]/[end]` lines are handled differently from the rest as they are conceptually different: not really drawing literals but instructions to `dvips` itself.

```

1031 \cs_new_protected:Npn \__draw_backend_begin:
1032 {
1033   \__kernel_backend_literal:n { ps::[begin] }
1034   \__draw_backend_literal:n { @beginspecial }
1035 }
1036 \cs_new_protected:Npn \__draw_backend_end:
1037 {
1038   \__draw_backend_literal:n { @endspecial }
1039   \__kernel_backend_literal:n { ps::[end] }
1040 }

```

(End definition for \\_\_draw\_backend\_begin: and \\_\_draw\_backend\_end:.)

Scope here may need to contain saved definitions, so the entire memory rather than just the graphic state has to be sent to the stack.

```

1041 \cs_new_protected:Npn \__draw_backend_scope_begin:
1042 { \__draw_backend_literal:n { save } }
1043 \cs_new_protected:Npn \__draw_backend_scope_end:
1044 { \__draw_backend_literal:n { restore } }

```

(End definition for `\_draw_backend_scope_begin:` and `\_draw_backend_scope_end:`.)

`\_draw_backend_moveto:nn` Path creation operations mainly resolve directly to PostScript primitive steps, with only the need to convert to bp. Notice that x-type expansion is included here to ensure that any variable values are forced to literals before any possible caching. There is no native rectangular path command (without also clipping, filling or stroking), so that task is done using a small amount of PostScript.

```

1045 \cs_new_protected:Npn \_draw_backend_moveto:nn #1#2
1046 {
1047   \_draw_backend_literal:x
1048   {
1049     \dim_to_decimal_in_bp:n {#1} ~
1050     \dim_to_decimal_in_bp:n {#2} ~ moveto
1051   }
1052 }
1053 \cs_new_protected:Npn \_draw_backend_lineto:nn #1#2
1054 {
1055   \_draw_backend_literal:x
1056   {
1057     \dim_to_decimal_in_bp:n {#1} ~
1058     \dim_to_decimal_in_bp:n {#2} ~ lineto
1059   }
1060 }
1061 \cs_new_protected:Npn \_draw_backend_rectangle:nnnn #1#2#3#4
1062 {
1063   \_draw_backend_literal:x
1064   {
1065     \dim_to_decimal_in_bp:n {#4} ~ \dim_to_decimal_in_bp:n {#3} ~
1066     \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~
1067     moveto~dup~0~rlineto~exch~0~exch~rlineto~neg~0~rlineto~closepath
1068   }
1069 }
1070 \cs_new_protected:Npn \_draw_backend_curveto:nnnnnn #1#2#3#4#5#6
1071 {
1072   \_draw_backend_literal:x
1073   {
1074     \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~
1075     \dim_to_decimal_in_bp:n {#3} ~ \dim_to_decimal_in_bp:n {#4} ~
1076     \dim_to_decimal_in_bp:n {#5} ~ \dim_to_decimal_in_bp:n {#6} ~
1077     curveto
1078   }
1079 }

```

(End definition for `\_draw_backend_moveto:nn` and others.)

`\_draw_backend_evenodd_rule:` The even-odd rule here can be implemented as a simply switch.

```

\g__draw_draw_eor_bool 1080 \cs_new_protected:Npn \_draw_backend_evenodd_rule:
1081 { \bool_gset_true:N \g__draw_draw_eor_bool }
1082 \cs_new_protected:Npn \_draw_backend_nonzero_rule:
1083 { \bool_gset_false:N \g__draw_draw_eor_bool }
1084 \bool_new:N \g__draw_draw_eor_bool

```

(End definition for `\_draw_backend_evenodd_rule:`, `\_draw_backend_nonzero_rule:`, and `\g__draw_draw_eor_bool`.)

`\__draw_backend_closepath:` Unlike PDF, PostScript doesn't track separate colors for strokes and other elements. It is  
`\__draw_backend_stroke:` also desirable to have the `clip` keyword after a stroke or fill. To achieve those outcomes,  
`\__draw_backend_closestroke:` there is some work to do. For color, the stroke color is simple but the fill one has to be  
`\__draw_backend_fill:` inserted by hand. For clipping, the required ordering is achieved using a T<sub>E</sub>X switch.  
`\__draw_backend_fillstroke:` All of the operations end with a new path instruction as they do not terminate (again in  
`\__draw_backend_clip:` contrast to PDF).  
`\__draw_backend_discardpath:`  
`\g__draw_draw_clip_bool`

```

1085 \cs_new_protected:Npn \__draw_backend_closepath:
1086 { \__draw_backend_literal:n { closepath } }
1087 \cs_new_protected:Npn \__draw_backend_stroke:
1088 {
1089   \__draw_backend_literal:n { gsave }
1090   \__draw_backend_literal:n { color.sc }
1091   \__draw_backend_literal:n { stroke }
1092   \__draw_backend_literal:n { grestore }
1093   \bool_if:NT \g__draw_draw_clip_bool
1094   {
1095     \__draw_backend_literal:x
1096     {
1097       \bool_if:NT \g__draw_draw_eor_bool { eo }
1098       clip
1099     }
1100   }
1101   \__draw_backend_literal:n { newpath }
1102   \bool_gset_false:N \g__draw_draw_clip_bool
1103 }
1104 \cs_new_protected:Npn \__draw_backend_closestroke:
1105 {
1106   \__draw_backend_closepath:
1107   \__draw_backend_stroke:
1108 }
1109 \cs_new_protected:Npn \__draw_backend_fill:
1110 {
1111   \__draw_backend_literal:n { gsave }
1112   \__draw_backend_literal:n { color.fc }
1113   \__draw_backend_literal:x
1114   {
1115     \bool_if:NT \g__draw_draw_eor_bool { eo }
1116     fill
1117   }
1118   \__draw_backend_literal:n { grestore }
1119   \bool_if:NT \g__draw_draw_clip_bool
1120   {
1121     \__draw_backend_literal:x
1122     {
1123       \bool_if:NT \g__draw_draw_eor_bool { eo }
1124       clip
1125     }
1126   }
1127   \__draw_backend_literal:n { newpath }
1128   \bool_gset_false:N \g__draw_draw_clip_bool
1129 }
1130 \cs_new_protected:Npn \__draw_backend_fillstroke:
1131 {

```

```

1132 \__draw_backend_literal:n { gsave }
1133 \__draw_backend_literal:n { color.sc }
1134 \__draw_backend_literal:n { color.fc }
1135 \__draw_backend_literal:x
1136 {
1137   \bool_if:NT \g__draw_draw_eor_bool { eo }
1138   fill
1139 }
1140 \__draw_backend_literal:n { grestore }
1141 \__draw_backend_literal:n { stroke }
1142 \bool_if:NT \g__draw_draw_clip_bool
1143 {
1144   \__draw_backend_literal:x
1145   {
1146     \bool_if:NT \g__draw_draw_eor_bool { eo }
1147     clip
1148   }
1149 }
1150 \__draw_backend_literal:n { newpath }
1151 \bool_gset_false:N \g__draw_draw_clip_bool
1152 }
1153 \cs_new_protected:Npn \__draw_backend_clip:
1154 { \bool_gset_true:N \g__draw_draw_clip_bool }
1155 \bool_new:N \g__draw_draw_clip_bool
1156 \cs_new_protected:Npn \__draw_backend_discardpath:
1157 {
1158   \bool_if:NT \g__draw_draw_clip_bool
1159   {
1160     \__draw_backend_literal:x
1161     {
1162       \bool_if:NT \g__draw_draw_eor_bool { eo }
1163       clip
1164     }
1165   }
1166   \__draw_backend_literal:n { newpath }
1167   \bool_gset_false:N \g__draw_draw_clip_bool
1168 }

```

(End definition for \\_\_draw\_backend\_closepath: and others.)

|                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> \__draw_backend_dash_pattern:nn \__draw_backend_dash:n \__draw_backend_linewidth:n \__draw_backend_miterlimit:n \__draw_backend_cap_butt: \__draw_backend_cap_round: \__draw_backend_cap_rectangle: \__draw_backend_join_miter: \__draw_backend_join_round: \__draw_backend_join_bevel: </pre> | <p>Converting paths to output is again a case of mapping directly to PostScript operations.</p> <pre> 1169 \cs_new_protected:Npn \__draw_backend_dash_pattern:nn #1#2 1170 { 1171   \__draw_backend_literal:x 1172   { 1173     [ 1174       \exp_args:Nf \use:n 1175       { \clist_map_function:nN {#1} \__draw_backend_dash:n } 1176     ] ~ 1177     \dim_to_decimal_in_bp:n {#2} ~ setdash 1178   } 1179 } 1180 \cs_new:Npn \__draw_backend_dash:n #1 1181 { ~ \dim_to_decimal_in_bp:n {#1} } </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

1182 \cs_new_protected:Npn \__draw_backend_linewidth:n #1
1183 {
1184   \__draw_backend_literal:n
1185     { \dim_to_decimal_in_bp:n {#1} ~ setlinewidth }
1186 }
1187 \cs_new_protected:Npn \__draw_backend_miterlimit:n #1
1188 { \__draw_backend_literal:n { #1 ~ setmiterlimit } }
1189 \cs_new_protected:Npn \__draw_backend_cap_but:
1190 { \__draw_backend_literal:n { 0 ~ setlinecap } }
1191 \cs_new_protected:Npn \__draw_backend_cap_round:
1192 { \__draw_backend_literal:n { 1 ~ setlinecap } }
1193 \cs_new_protected:Npn \__draw_backend_cap_rectangle:
1194 { \__draw_backend_literal:n { 2 ~ setlinecap } }
1195 \cs_new_protected:Npn \__draw_backend_join_miter:
1196 { \__draw_backend_literal:n { 0 ~ setlinejoin } }
1197 \cs_new_protected:Npn \__draw_backend_join_round:
1198 { \__draw_backend_literal:n { 1 ~ setlinejoin } }
1199 \cs_new_protected:Npn \__draw_backend_join_bevel:
1200 { \__draw_backend_literal:n { 2 ~ setlinejoin } }

```

(End definition for \\_\_draw\_backend\_dash\_pattern:nn and others.)

\\_\_draw\_backend\_cm:nnnn

In dvips, keeping the transformations in line with the engine is unfortunately not possible for scaling and rotations: even if we decompose the matrix into those operations, there is still no backend tracking (cf. dvipdfmx/X<sub>Y</sub>TeX). Thus we take the shortest path available and simply dump the matrix as given.

```

1201 \cs_new_protected:Npn \__draw_backend_cm:nnnn #1#2#3#4
1202 {
1203   \__draw_backend_literal:n
1204     { [ #1 ~ #2 ~ #3 ~ #4 ~ 0 ~ 0 ] ~ concat }
1205 }

```

(End definition for \\_\_draw\_backend\_cm:nnnn.)

\\_\_draw\_backend\_box\_use:Nnnnn

Inside a picture @beginspecial/@endspecial are active, which is normally a good thing but means that the position and scaling would be off if the box was inserted directly. To deal with that, there are a number of possible approaches. The implementation here was suggested by Tom Rokici (author of dvips). We end the current special placement, then set the current point with a literal [begin]. As for general literals, we then use the stack to store the current point and move to it. To insert the required transformation, we have to flip the  $y$ -axis, once before and once after it. Then we get back to the T<sub>E</sub>X reference point to insert our content. The clean up has to happen in the right places, hence the [begin]/[end] pair around restore. Finally, we can return to “normal” drawing mode. Notice that the set up here is very similar to that in \\_\_draw\_align\_currentpoint..., but the ordering of saving and restoring is different (intermixed).

```

1206 \cs_new_protected:Npn \__draw_backend_box_use:Nnnnn #1#2#3#4#5
1207 {
1208   \__draw_backend_literal:n { @endspecial }
1209   \__draw_backend_literal:n { [end] }
1210   \__draw_backend_literal:n { [begin] }
1211   \__draw_backend_literal:n { save }
1212   \__draw_backend_literal:n { currentpoint }
1213   \__draw_backend_literal:n { currentpoint~translate }

```



```

1214 \__draw_backend_cm:nnnn { 1 } { 0 } { 0 } { -1 }
1215 \__draw_backend_cm:nnnn {#2} {#3} {#4} {#5}
1216 \__draw_backend_cm:nnnn { 1 } { 0 } { 0 } { -1 }
1217 \__draw_backend_literal:n { neg~exch~neg~exch~translate }
1218 \__draw_backend_literal:n { [end] }
1219 \hbox_overlap_right:n { \box_use:N #1 }
1220 \__draw_backend_literal:n { [begin] }
1221 \__draw_backend_literal:n { restore }
1222 \__draw_backend_literal:n { [end] }
1223 \__draw_backend_literal:n { [begin] }
1224 \__draw_backend_literal:n { @beginspecial }
1225 }

```

(End definition for \\_\_draw\_backend\_box\_use:Nnnnn.)

```

1226 </dvips>

```

## 4.2 LuaTeX, pdfTeX, dvipdfmx and XeTeX

LuaTeX, pdfTeX, dvipdfmx and XeTeX directly produce PDF output and understand a shared set of specials for drawing commands.

```

1227 <*dvipdfmx | luatex | pdftex | xetex>

```

### 4.2.1 Drawing

```

\__draw_backend_literal:n Pass data through using a dedicated interface.
\__draw_backend_literal:x
1228 \cs_new_eq:NN \__draw_backend_literal:n \__kernel_backend_literal_pdf:n
1229 \cs_generate_variant:Nn \__draw_backend_literal:n { x }

```

(End definition for \\_\_draw\_backend\_literal:n.)

```

\__draw_backend_begin: No special requirements here, so simply set up a drawing scope.
\__draw_backend_end:
1230 \cs_new_protected:Npn \__draw_backend_begin:
1231 { \__draw_backend_scope_begin: }
1232 \cs_new_protected:Npn \__draw_backend_end:
1233 { \__draw_backend_scope_end: }

```

(End definition for \\_\_draw\_backend\_begin: and \\_\_draw\_backend\_end:.)

```

\__draw_backend_scope_begin: Use the backend-level scope mechanisms.
\__draw_backend_scope_end:
1234 \cs_new_eq:NN \__draw_backend_scope_begin: \__kernel_backend_scope_begin:
1235 \cs_new_eq:NN \__draw_backend_scope_end: \__kernel_backend_scope_end:

```

(End definition for \\_\_draw\_backend\_scope\_begin: and \\_\_draw\_backend\_scope\_end:.)

```

\__draw_backend_moveto:nn Path creation operations all resolve directly to PDF primitive steps, with only the need
\__draw_backend_lineto:nn to convert to bp.
\__draw_backend_curveto:nnnnn
\__draw_backend_rectangle:nnnn
1236 \cs_new_protected:Npn \__draw_backend_moveto:nn #1#2
1237 {
1238 \__draw_backend_literal:x
1239 { \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~ m }
1240 }
1241 \cs_new_protected:Npn \__draw_backend_lineto:nn #1#2
1242 {
1243 \__draw_backend_literal:x

```

```

1244     { \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~ 1 }
1245   }
1246 \cs_new_protected:Npn \__draw_backend_curveto:nnnnn #1#2#3#4#5#6
1247 {
1248   \__draw_backend_literal:x
1249   {
1250     \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~
1251     \dim_to_decimal_in_bp:n {#3} ~ \dim_to_decimal_in_bp:n {#4} ~
1252     \dim_to_decimal_in_bp:n {#5} ~ \dim_to_decimal_in_bp:n {#6} ~
1253     c
1254   }
1255 }
1256 \cs_new_protected:Npn \__draw_backend_rectangle:nnnn #1#2#3#4
1257 {
1258   \__draw_backend_literal:x
1259   {
1260     \dim_to_decimal_in_bp:n {#1} ~ \dim_to_decimal_in_bp:n {#2} ~
1261     \dim_to_decimal_in_bp:n {#3} ~ \dim_to_decimal_in_bp:n {#4} ~
1262     re
1263   }
1264 }

```

(End definition for \\_\_draw\_backend\_moveto:nn and others.)

\\_\_draw\_backend\_evenodd\_rule: The even-odd rule here can be implemented as a simply switch.

```

1265 \cs_new_protected:Npn \__draw_backend_evenodd_rule:
1266   { \bool_gset_true:N \g__draw_draw_eor_bool }
1267 \cs_new_protected:Npn \__draw_backend_nonzero_rule:
1268   { \bool_gset_false:N \g__draw_draw_eor_bool }
1269 \bool_new:N \g__draw_draw_eor_bool

```

(End definition for \\_\_draw\_backend\_evenodd\_rule:, \\_\_draw\_backend\_nonzero\_rule:, and \g\_\_draw\_draw\_eor\_bool.)

\\_\_draw\_backend\_closepath: Converting paths to output is again a case of mapping directly to PDF operations.

```

1270 \cs_new_protected:Npn \__draw_backend_closepath:
1271   { \__draw_backend_literal:n { h } }
1272 \cs_new_protected:Npn \__draw_backend_stroke:
1273   { \__draw_backend_literal:n { S } }
1274 \cs_new_protected:Npn \__draw_backend_closestroke:
1275   { \__draw_backend_literal:n { s } }
1276 \cs_new_protected:Npn \__draw_backend_fill:
1277   {
1278     \__draw_backend_literal:x
1279     { f \bool_if:NT \g__draw_draw_eor_bool * }
1280   }
1281 \cs_new_protected:Npn \__draw_backend_fillstroke:
1282   {
1283     \__draw_backend_literal:x
1284     { B \bool_if:NT \g__draw_draw_eor_bool * }
1285   }
1286 \cs_new_protected:Npn \__draw_backend_clip:
1287   {
1288     \__draw_backend_literal:x
1289     { W \bool_if:NT \g__draw_draw_eor_bool * }

```

```

1290 }
1291 \cs_new_protected:Npn \__draw_backend_discardpath:
1292 { \__draw_backend_literal:n { n } }

```

(End definition for \\_\_draw\_backend\_closepath: and others.)

Converting paths to output is again a case of mapping directly to PDF operations.

```

\__draw_backend_dash_pattern:nn
\__draw_backend_dash:n
\__draw_backend_linewidth:n
\__draw_backend_miterlimit:n
\__draw_backend_cap_but:
\__draw_backend_cap_round:
\__draw_backend_cap_rectangle:
\__draw_backend_join_miter:
\__draw_backend_join_round:
\__draw_backend_join_bevel:
1293 \cs_new_protected:Npn \__draw_backend_dash_pattern:nn #1#2
1294 {
1295   \__draw_backend_literal:x
1296   {
1297     [
1298       \exp_args:Nf \use:n
1299       { \clist_map_function:nN {#1} \__draw_backend_dash:n }
1300     ] ~
1301     \dim_to_decimal_in_bp:n {#2} ~ d
1302   }
1303 }
1304 \cs_new:Npn \__draw_backend_dash:n #1
1305 { ~ \dim_to_decimal_in_bp:n {#1} }
1306 \cs_new_protected:Npn \__draw_backend_linewidth:n #1
1307 {
1308   \__draw_backend_literal:x
1309   { \dim_to_decimal_in_bp:n {#1} ~ w }
1310 }
1311 \cs_new_protected:Npn \__draw_backend_miterlimit:n #1
1312 { \__draw_backend_literal:x { #1 ~ M } }
1313 \cs_new_protected:Npn \__draw_backend_cap_but:
1314 { \__draw_backend_literal:n { 0 ~ J } }
1315 \cs_new_protected:Npn \__draw_backend_cap_round:
1316 { \__draw_backend_literal:n { 1 ~ J } }
1317 \cs_new_protected:Npn \__draw_backend_cap_rectangle:
1318 { \__draw_backend_literal:n { 2 ~ J } }
1319 \cs_new_protected:Npn \__draw_backend_join_miter:
1320 { \__draw_backend_literal:n { 0 ~ j } }
1321 \cs_new_protected:Npn \__draw_backend_join_round:
1322 { \__draw_backend_literal:n { 1 ~ j } }
1323 \cs_new_protected:Npn \__draw_backend_join_bevel:
1324 { \__draw_backend_literal:n { 2 ~ j } }

```

(End definition for \\_\_draw\_backend\_dash\_pattern:nn and others.)

Another split here between LuaTeX/pdfTeX and dvipdfmx/X<sub>Y</sub>TeX. In the former, we have a direct method to maintain alignment: the backend can use a matrix itself. For dvipdfmx/X<sub>Y</sub>TeX, we can to decompose the matrix into rotations and a scaling, then use those operations as they are handled by the backend. (There is backend support for matrix operations in dvipdfmx/X<sub>Y</sub>TeX, but as a matched pair so not suitable for the “stand alone” transformation set up here.) The specials used here are from xdvipdfmx originally: they are well-tested, but probably equivalent to the pdf<sub>+</sub> versions!

```

1325 \cs_new_protected:Npn \__draw_backend_cm:nnnn #1#2#3#4
1326 {
1327   <*luatex | pdftex>
1328   \__kernel_backend_matrix:n { #1 ~ #2 ~ #3 ~ #4 }
1329   </luatex | pdftex>

```

```

1330 <*dvipdfmx | xetex>
1331   \_draw_backend_cm_decompose:nnnnN {#1} {#2} {#3} {#4}
1332   \_draw_backend_cm_aux:nnnn
1333 </dvipdfmx | xetex>
1334 }
1335 <*dvipdfmx | xetex>
1336 \cs_new_protected:Npn \_draw_backend_cm_aux:nnnn #1#2#3#4
1337 {
1338   \_kernel_backend_literal:x
1339   {
1340     x:rotate~
1341     \fp_compare:nNnTF {#1} = \c_zero_fp
1342     { 0 }
1343     { \fp_eval:n { round ( -#1 , 5 ) } }
1344   }
1345   \_kernel_backend_literal:x
1346   {
1347     x:scale~
1348     \fp_eval:n { round ( #2 , 5 ) } ~
1349     \fp_eval:n { round ( #3 , 5 ) }
1350   }
1351   \_kernel_backend_literal:x
1352   {
1353     x:rotate~
1354     \fp_compare:nNnTF {#4} = \c_zero_fp
1355     { 0 }
1356     { \fp_eval:n { round ( -#4 , 5 ) } }
1357   }
1358 }
1359 </dvipdfmx | xetex>

```

(End definition for `\_draw_backend_cm:nnnn` and `\_draw_backend_cm_aux:nnnn`.)

```

\_draw_backend_cm_decompose:nnnnN
\_draw_backend_cm_decompose_auxi:nnnnN
\_draw_backend_cm_decompose_auxii:nnnnN
\_draw_backend_cm_decompose_auxiii:nnnnN

```

Internally, transformations for drawing are tracked as a matrix. Not all engines provide a way of dealing with this: if we use a raw matrix, the engine loses track of positions (for example for hyperlinks), and this is not desirable. They do, however, allow us to track rotations and scalings. Luckily, we can decompose any (two-dimensional) matrix into two rotations and a single scaling:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix} \begin{bmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{bmatrix}$$

The parent matrix can be converted to

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} E & H \\ -H & E \end{bmatrix} + \begin{bmatrix} F & G \\ G & -F \end{bmatrix}$$

From these, we can find that

$$\begin{aligned} \frac{w_1 + w_2}{2} &= \sqrt{E^2 + H^2} \\ \frac{w_1 - w_2}{2} &= \sqrt{F^2 + G^2} \\ \gamma - \beta &= \tan^{-1}(G/F) \\ \gamma + \beta &= \tan^{-1}(H/E) \end{aligned}$$

at which point we just have to do various pieces of re-arrangement to get all of the values. (See J. Blinn, *IEEE Comput. Graph. Appl.*, 1996, **16**, 82–88.) There is one wrinkle: the PostScript (and PDF) way of specifying a transformation matrix exchanges where one would normally expect  $B$  and  $C$  to be.

```

1360 <*dvipdfmx|xetex>
1361 \cs_new_protected:Npn \__draw_backend_cm_decompose:nnnnN #1#2#3#4#5
1362 {
1363   \use:x
1364   {
1365     \__draw_backend_cm_decompose_auxi:nnnnN
1366     { \fp_eval:n { (#1 + #4) / 2 } }
1367     { \fp_eval:n { (#1 - #4) / 2 } }
1368     { \fp_eval:n { (#3 + #2) / 2 } }
1369     { \fp_eval:n { (#3 - #2) / 2 } }
1370   }
1371   #5
1372 }
1373 \cs_new_protected:Npn \__draw_backend_cm_decompose_auxi:nnnnN #1#2#3#4#5
1374 {
1375   \use:x
1376   {
1377     \__draw_backend_cm_decompose_auxii:nnnnN
1378     { \fp_eval:n { 2 * sqrt ( #1 * #1 + #4 * #4 ) } }
1379     { \fp_eval:n { 2 * sqrt ( #2 * #2 + #3 * #3 ) } }
1380     { \fp_eval:n { atand ( #3 , #2 ) } }
1381     { \fp_eval:n { atand ( #4 , #1 ) } }
1382   }
1383   #5
1384 }
1385 \cs_new_protected:Npn \__draw_backend_cm_decompose_auxii:nnnnN #1#2#3#4#5
1386 {
1387   \use:x
1388   {
1389     \__draw_backend_cm_decompose_auxiii:nnnnN
1390     { \fp_eval:n { ( #4 - #3 ) / 2 } }
1391     { \fp_eval:n { ( #1 + #2 ) / 2 } }
1392     { \fp_eval:n { ( #1 - #2 ) / 2 } }
1393     { \fp_eval:n { ( #4 + #3 ) / 2 } }
1394   }
1395   #5
1396 }
1397 \cs_new_protected:Npn \__draw_backend_cm_decompose_auxiii:nnnnN #1#2#3#4#5
1398 {
1399   \fp_compare:nNnTF { abs ( #2 ) } > { abs ( #3 ) }
1400   { #5 {#1} {#2} {#3} {#4} }
1401   { #5 {#1} {#3} {#2} {#4} }
1402 }
1403 </dvipdfmx|xetex>

```

(End definition for `\__draw_backend_cm_decompose:nnnnN` and others.)

`\_draw_backend_box_use:Nnnnn`

Inserting a T<sub>E</sub>X box transformed to the requested position and using the current matrix is done using a mixture of T<sub>E</sub>X and low-level manipulation. The offset can be handled by T<sub>E</sub>X, so only any rotation/skew/scaling component needs to be done using the matrix

operation. As this operation can never be cached, the scope is set directly not using the `draw` version.

```

1404 \cs_new_protected:Npn \__draw_backend_box_use:Nnnnn #1#2#3#4#5
1405 {
1406   \__kernel_backend_scope_begin:
1407   <*luatex | pdftex>
1408   \__draw_backend_cm:nnnn {#2} {#3} {#4} {#5}
1409   </luatex | pdftex>
1410   <*dvipdfmx | xetex>
1411   \__kernel_backend_literal:n
1412   { pdf:btrans~matrix~ #2 ~ #3 ~ #4 ~ #5 ~ 0 ~ 0 }
1413   </dvipdfmx | xetex>
1414   \hbox_overlap_right:n { \box_use:N #1 }
1415   <*dvipdfmx | xetex>
1416   \__kernel_backend_literal:n { pdf:etrans }
1417   </dvipdfmx | xetex>
1418   \__kernel_backend_scope_end:
1419 }
(End definition for \__draw_backend_box_use:Nnnnn.)
1420 </dvipdfmx | luatex | pdftex | xetex>

```

### 4.3 dvisvgm backend

```

1421 <*dvisvgm>

```

The same as the more general literal call.

```

1422 \cs_new_eq:NN \__draw_backend_literal:n \__kernel_backend_literal_svg:n
1423 \cs_generate_variant:Nn \__draw_backend_literal:n { x }

```

(End definition for `\__draw_backend_literal:n`.)

`\__draw_backend_begin:` A drawing needs to be set up such that the co-ordinate system is translated. That is done inside a scope, which as described below

```

1424 \cs_new_protected:Npn \__draw_backend_begin:
1425 {
1426   \__kernel_backend_scope_begin:
1427   \__kernel_backend_scope:n { transform="translate({?x},{?y})~scale(1,-1)" }
1428 }
1429 \cs_new_eq:NN \__draw_backend_end: \__kernel_backend_scope_end:

```

(End definition for `\__draw_backend_begin:` and `\__draw_backend_end:.`)

`\__draw_backend_moveto:nn` Once again, some work is needed to get path constructs correct. Rather than write the values as they are given, the entire path needs to be collected up before being output in one go. For that we use a dedicated storage routine, which adds spaces as required. `\__draw_backend_lineto:nn` Since paths should be fully expanded there is no need to worry about the internal x-type expansion.

```

1430 \cs_new_protected:Npn \__draw_backend_moveto:nn #1#2
1431 {
1432   \__draw_backend_add_to_path:n
1433   { M ~ \dim_to_decimal:n {#1} ~ \dim_to_decimal:n {#2} }
1434 }
1435 \cs_new_protected:Npn \__draw_backend_lineto:nn #1#2

```

```

1436 {
1437   \__draw_backend_add_to_path:n
1438   { L ~ \dim_to_decimal:n {#1} ~ \dim_to_decimal:n {#2} }
1439 }
1440 \cs_new_protected:Npn \__draw_backend_rectangle:nnnn #1#2#3#4
1441 {
1442   \__draw_backend_add_to_path:n
1443   {
1444     M ~ \dim_to_decimal:n {#1} ~ \dim_to_decimal:n {#2}
1445     h ~ \dim_to_decimal:n {#3} ~
1446     v ~ \dim_to_decimal:n {#4} ~
1447     h ~ \dim_to_decimal:n { -#3 } ~
1448     Z
1449   }
1450 }
1451 \cs_new_protected:Npn \__draw_backend_curveto:nnnnnn #1#2#3#4#5#6
1452 {
1453   \__draw_backend_add_to_path:n
1454   {
1455     C ~
1456     \dim_to_decimal:n {#1} ~ \dim_to_decimal:n {#2} ~
1457     \dim_to_decimal:n {#3} ~ \dim_to_decimal:n {#4} ~
1458     \dim_to_decimal:n {#5} ~ \dim_to_decimal:n {#6}
1459   }
1460 }
1461 \cs_new_protected:Npn \__draw_backend_add_to_path:n #1
1462 {
1463   \tl_gset:Nx \g__draw_draw_path_tl
1464   {
1465     \g__draw_draw_path_tl
1466     \tl_if_empty:NF \g__draw_draw_path_tl { \c_space_tl }
1467     #1
1468   }
1469 }
1470 \tl_new:N \g__draw_draw_path_tl

```

(End definition for \\_\_draw\_backend\_moveto:nn and others.)

```

\__draw_backend_evenodd_rule: The fill rules here have to be handled as scopes.
\__draw_backend_nonzero_rule: 1471 \cs_new_protected:Npn \__draw_backend_evenodd_rule:
                                1472 { \__draw_backend_scope:n { fill-rule="evenodd" } }
                                1473 \cs_new_protected:Npn \__draw_backend_nonzero_rule:
                                1474 { \__draw_backend_scope:n { fill-rule="nonzero" } }

```

(End definition for \\_\_draw\_backend\_evenodd\_rule: and \\_\_draw\_backend\_nonzero\_rule:.)

```

\__draw_backend_path:n Setting fill and stroke effects and doing clipping all has to be done using scopes. This
\__draw_backend_closepath: means setting up the various requirements in a shared auxiliary which deals with the
\__draw_backend_stroke: bits and pieces. Clipping paths are reused for path drawing; not essential but avoids
\__draw_backend_closestroke: constructing them twice. Discarding a path needs a separate function as it's not quite
\__draw_backend_fill: the same.
\__draw_backend_fillstroke: 1475 \cs_new_protected:Npn \__draw_backend_closepath:
\__draw_backend_clip: 1476 { \__draw_backend_add_to_path:n { Z } }
\__draw_backend_discardpath: 1477 \cs_new_protected:Npn \__draw_backend_path:n #1
\g__draw_draw_clip_bool
\g__draw_draw_path_int

```

```

1478 {
1479     \bool_if:NTF \g__draw_draw_clip_bool
1480     {
1481         \int_gincr:N \g__draw_clip_path_int
1482         \__draw_backend_literal:x
1483         {
1484             < clipPath~id = " l3cp \int_use:N \g__draw_clip_path_int " >
1485             { ?nl }
1486             <path~d=" \g__draw_draw_path_tl "/> { ?nl }
1487             < /clipPath > { ? nl }
1488             <
1489                 use~xlink:href =
1490                 " \c_hash_str l3path \int_use:N \g__draw_path_int " ~
1491                 #1
1492             />
1493         }
1494         \__draw_backend_scope:x
1495         {
1496             clip-path =
1497             "url( \c_hash_str l3cp \int_use:N \g__draw_clip_path_int )"
1498         }
1499     }
1500     {
1501         \__draw_backend_literal:x
1502         { <path ~ d=" \g__draw_draw_path_tl " ~ #1 /> }
1503     }
1504     \tl_gclear:N \g__draw_draw_path_tl
1505     \bool_gset_false:N \g__draw_draw_clip_bool
1506 }
1507 \int_new:N \g__draw_path_int
1508 \cs_new_protected:Npn \__draw_backend_stroke:
1509 { \__draw_backend_path:n { style="fill:none" } }
1510 \cs_new_protected:Npn \__draw_backend_closestroke:
1511 {
1512     \__draw_backend_closepath:
1513     \__draw_backend_stroke:
1514 }
1515 \cs_new_protected:Npn \__draw_backend_fill:
1516 { \__draw_backend_path:n { style="stroke:none" } }
1517 \cs_new_protected:Npn \__draw_backend_fillstroke:
1518 { \__draw_backend_path:n { } }
1519 \cs_new_protected:Npn \__draw_backend_clip:
1520 { \bool_gset_true:N \g__draw_draw_clip_bool }
1521 \bool_new:N \g__draw_draw_clip_bool
1522 \cs_new_protected:Npn \__draw_backend_discardpath:
1523 {
1524     \bool_if:NT \g__draw_draw_clip_bool
1525     {
1526         \int_gincr:N \g__draw_clip_path_int
1527         \__draw_backend_literal:x
1528         {
1529             < clipPath~id = " l3cp \int_use:N \g__draw_clip_path_int " >
1530             { ?nl }
1531             <path~d=" \g__draw_draw_path_tl "/> { ?nl }

```



```

1532         < /clipPath >
1533     }
1534     \__draw_backend_scope:x
1535     {
1536         clip-path =
1537             "url( \c_hash_str l3cp \int_use:N \g__draw_clip_path_int)"
1538     }
1539 }
1540 \tl_gclear:N \g__draw_draw_path_tl
1541 \bool_gset_false:N \g__draw_draw_clip_bool
1542 }

```

(End definition for \\_\_draw\_backend\_path:n and others.)

All of these ideas are properties of scopes in SVG. The only slight complexity is converting the dash array properly (doing any required maths).

```

\__draw_backend_dash_pattern:nn
\__draw_backend_dash:n
\__draw_backend_dash_aux:nn
\__draw_backend_linewidth:n
\__draw_backend_miterlimit:n
\__draw_backend_cap_but:
\__draw_backend_cap_round:
\__draw_backend_cap_rectangle:
\__draw_backend_join_miter:
\__draw_backend_join_round:
\__draw_backend_join_bevel:
1543 \cs_new_protected:Npn \__draw_backend_dash_pattern:nn #1#2
1544 {
1545     \use:x
1546     {
1547         \__draw_backend_dash_aux:nn
1548         { \clist_map_function:nn {#1} \__draw_backend_dash:n }
1549         { \dim_to_decimal:n {#2} }
1550     }
1551 }
1552 \cs_new:Npn \__draw_backend_dash:n #1
1553 { , \dim_to_decimal_in_bp:n {#1} }
1554 \cs_new_protected:Npn \__draw_backend_dash_aux:nn #1#2
1555 {
1556     \__draw_backend_scope:x
1557     {
1558         stroke-dasharray =
1559         "
1560         \tl_if_empty:oTF { \use_none:n #1 }
1561         { none }
1562         { \use_none:n #1 }
1563         " ~
1564         stroke-offset=" #2 "
1565     }
1566 }
1567 \cs_new_protected:Npn \__draw_backend_linewidth:n #1
1568 { \__draw_backend_scope:x { stroke-width=" \dim_to_decimal:n {#1} " } }
1569 \cs_new_protected:Npn \__draw_backend_miterlimit:n #1
1570 { \__draw_backend_scope:x { stroke-miterlimit=" #1 " } }
1571 \cs_new_protected:Npn \__draw_backend_cap_but:
1572 { \__draw_backend_scope:n { stroke-linecap="butt" } }
1573 \cs_new_protected:Npn \__draw_backend_cap_round:
1574 { \__draw_backend_scope:n { stroke-linecap="round" } }
1575 \cs_new_protected:Npn \__draw_backend_cap_rectangle:
1576 { \__draw_backend_scope:n { stroke-linecap="square" } }
1577 \cs_new_protected:Npn \__draw_backend_join_miter:
1578 { \__draw_backend_scope:n { stroke-linejoin="miter" } }
1579 \cs_new_protected:Npn \__draw_backend_join_round:
1580 { \__draw_backend_scope:n { stroke-linejoin="round" } }

```

```

1581 \cs_new_protected:Npn \__draw_backend_join_bevel:
1582 { \__draw_backend_scope:n { stroke-linejoin="bevel" } }

```

(End definition for \\_\_draw\_backend\_dash\_pattern:nn and others.)

\\_\_draw\_backend\_cm:nnnn The four arguments here are floats (the affine matrix), the last two are a displacement vector.

```

1583 \cs_new_protected:Npn \__draw_backend_cm:nnnn #1#2#3#4
1584 {
1585   \__draw_backend_scope:n
1586   {
1587     transform =
1588     " matrix ( #1 , #2 , #3 , #4 , Opt , Opt ) "
1589   }
1590 }

```

(End definition for \\_\_draw\_backend\_cm:nnnn.)

\\_\_draw\_backend\_box\_use:Nnnnn No special savings can be made here: simply displace the box inside a scope. As there is nothing to re-box, just make the box passed of zero size.

```

1591 \cs_new_protected:Npn \__draw_backend_box_use:Nnnnn #1#2#3#4#5#6#7
1592 {
1593   \__kernel_backend_scope_begin:
1594   \__draw_backend_cm:nnnn {#2} {#3} {#4} {#5}
1595   \__kernel_backend_literal_svg:n
1596   {
1597     < g~
1598     stroke="none"~
1599     transform="scale(-1,1)~translate({?x},{?y})~scale(-1,-1)"
1600     >
1601   }
1602   \box_set_wd:Nn #1 { Opt }
1603   \box_set_ht:Nn #1 { Opt }
1604   \box_set_dp:Nn #1 { Opt }
1605   \box_use:N #1
1606   \__kernel_backend_literal_svg:n { </g> }
1607   \__kernel_backend_scope_end:
1608 }

```

(End definition for \\_\_draw\_backend\_box\_use:Nnnnn.)

```

1609 </dvisvgm>

```

```

1610 </package>

```

## 5 l3backend-graphics Implementation

```

1611 <*package>
1612 <@@=graphics>

```

### 5.1 dvips backend

```

1613 <*dvips>

```

\\_graphics\_backend\_getbb\_eps:n Simply use the generic function.

```

1614 \cs_new_eq:NN \_graphics_backend_getbb_eps:n \graphics_read_bb:n

```

(End definition for `\_graphics_backend_getbb_eps:n`.)

`\_graphics_backend_include_eps:n` The special syntax is relatively clear here: remember we need PostScript sizes here.

```

1615 \cs_new_protected:Npn \_graphics_backend_include_eps:n #1
1616 {
1617   \_kernel_backend_literal:x
1618   {
1619     PSfile = #1 \c_space_tl
1620     llx = \dim_to_decimal_in_bp:n \l_graphics_llx_dim \c_space_tl
1621     lly = \dim_to_decimal_in_bp:n \l_graphics_lly_dim \c_space_tl
1622     urx = \dim_to_decimal_in_bp:n \l_graphics_urx_dim \c_space_tl
1623     ury = \dim_to_decimal_in_bp:n \l_graphics_ury_dim
1624   }
1625 }

```

(End definition for `\_graphics_backend_include_eps:n`.)

```

1626 </dvips>

```

## 5.2 LuaTeX and pdfTeX backends

```

1627 <*\luatex | pdftex>

```

`\l_graphics_graphics_attr_tl` In PDF mode, additional attributes of an graphic (such as page number) are needed both to obtain the bounding box and when inserting the graphic: this occurs as the graphic dictionary approach means they are read as part of the bounding box operation. As such, it is easier to track additional attributes using a dedicated `tl` rather than build up the same data twice.

```

1628 \tl_new:N \l_graphics_graphics_attr_tl

```

(End definition for `\l_graphics_graphics_attr_tl`.)

`\_graphics_backend_getbb_jpg:n`  
`\_graphics_backend_getbb_pdf:n`  
`\_graphics_backend_getbb_png:n`  
`\_graphics_backend_getbb_auxi:n`  
`\_graphics_backend_getbb_auxii:n` Getting the bounding box here requires us to box up the graphic and measure it. To deal with the difference in feature support in bitmap and vector graphics but keeping the common parts, there is a little work to do in terms of auxiliaries. The key here is to notice that we need two forms of the attributes: a “short” set to allow us to track for caching, and the full form to pass to the primitive.

```

1629 \cs_new_protected:Npn \_graphics_backend_getbb_jpg:n #1
1630 {
1631   \int_zero:N \l_graphics_page_int
1632   \tl_clear:N \l_graphics_pagebox_tl
1633   \tl_set:Nx \l_graphics_graphics_attr_tl
1634   {
1635     \tl_if_empty:NF \l_graphics_decodearray_tl
1636     { :D \l_graphics_decodearray_tl }
1637     \bool_if:NT \l_graphics_interpolate_bool
1638     { :I }
1639   }
1640   \tl_clear:N \l_graphics_graphics_attr_tl
1641   \_graphics_backend_getbb_auxi:n {#1}
1642 }
1643 \cs_new_eq:NN \_graphics_backend_getbb_png:n \_graphics_backend_getbb_jpg:n
1644 \cs_new_protected:Npn \_graphics_backend_getbb_pdf:n #1
1645 {

```

```

1646 \tl_clear:N \l_graphics_decodearray_tl
1647 \bool_set_false:N \l_graphics_interpolate_bool
1648 \tl_set:Nx \l__graphics_graphics_attr_tl
1649 {
1650   : \l_graphics_pagebox_tl
1651   \int_compare:nNnT \l_graphics_page_int > 1
1652     { :P \int_use:N \l_graphics_page_int }
1653 }
1654 \__graphics_backend_getbb_auxi:n {#1}
1655 }
1656 \cs_new_protected:Npn \__graphics_backend_getbb_auxi:n #1
1657 {
1658   \graphics_bb_restore:xF { #1 \l__graphics_graphics_attr_tl }
1659   { \__graphics_backend_getbb_auxii:n {#1} }
1660 }

```

Measuring the graphic is done by boxing up: for PDF graphics we could use `\tex_pdfximagebbox:D`, but if doesn't work for other types. As the box always starts at (0,0) there is no need to worry about the lower-left position.

```

1661 \cs_new_protected:Npn \__graphics_backend_getbb_auxii:n #1
1662 {
1663   \tex_immediate:D \tex_pdfximage:D
1664   \bool_lazy_or:nnT
1665     { \l_graphics_interpolate_bool }
1666     { ! \tl_if_empty_p:N \l_graphics_decodearray_tl }
1667   {
1668     attr ~
1669     {
1670       \tl_if_empty:NF \l_graphics_decodearray_tl
1671       { /Decode~[ \l_graphics_decodearray_tl ] }
1672       \bool_if:NT \l_graphics_interpolate_bool
1673       { /Interpolate~true }
1674     }
1675   }
1676   \int_compare:nNnT \l_graphics_page_int > 0
1677     { page ~ \int_use:N \l_graphics_page_int }
1678   \tl_if_empty:NF \l_graphics_pagebox_tl
1679     { \l_graphics_pagebox_tl }
1680   {#1}
1681   \hbox_set:Nn \l__graphics_internal_box
1682     { \tex_pdfrefximage:D \tex_pdflastximage:D }
1683   \dim_set:Nn \l_graphics_urx_dim { \box_wd:N \l__graphics_internal_box }
1684   \dim_set:Nn \l_graphics_ury_dim { \box_ht:N \l__graphics_internal_box }
1685   \int_const:cn { c__graphics_graphics_ #1 \l__graphics_graphics_attr_tl _int }
1686     { \tex_the:D \tex_pdflastximage:D }
1687   \graphics_bb_save:x { #1 \l__graphics_graphics_attr_tl }
1688 }

```

(End definition for `\__graphics_backend_getbb_jpg:n` and others.)

```

\_graphics_backend_include_jpg:n
\_graphics_backend_include_pdf:n
\_graphics_backend_include_png:n

```

Images are already loaded for the measurement part of the code, so inclusion is straightforward, with only any attributes to worry about. The latter carry through from determination of the bounding box.

```

1689 \cs_new_protected:Npn \__graphics_backend_include_jpg:n #1

```

```

1690 {
1691   \tex_pdfrefximage:D
1692   \int_use:c { c__graphics_graphics_ #1 \l__graphics_graphics_attr_tl _int }
1693 }
1694 \cs_new_eq:NN \__graphics_backend_include_pdf:n \__graphics_backend_include_jpg:n
1695 \cs_new_eq:NN \__graphics_backend_include_png:n \__graphics_backend_include_jpg:n

```

(End definition for \\_\_graphics\_backend\_include\_jpg:n, \\_\_graphics\_backend\_include\_pdf:n, and \\_\_graphics\_backend\_include\_png:n.)

EPS graphics may be included in LuaTeX/pdfTeX by conversion to PDF: this requires restricted shell escape. Modelled on the `epstopdf` L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> package, but simplified, conversion takes place here if we have shell access.

```

\__graphics_backend_getbb_eps:n
\__graphics_backend_getbb_eps:nm
\__graphics_backend_include_eps:n
\l__graphics_backend_dir_str
\l__graphics_backend_name_str
\l__graphics_backend_ext_str
1696 \sys_if_shell:T
1697 {
1698   \str_new:N \l__graphics_backend_dir_str
1699   \str_new:N \l__graphics_backend_name_str
1700   \str_new:N \l__graphics_backend_ext_str
1701   \cs_new_protected:Npn \__graphics_backend_getbb_eps:n #1
1702   {
1703     \file_parse_full_name:nNNN {#1}
1704     \l__graphics_backend_dir_str
1705     \l__graphics_backend_name_str
1706     \l__graphics_backend_ext_str
1707     \exp_args:Nx \__graphics_backend_getbb_eps:nn
1708     {
1709       \l__graphics_backend_name_str - \str_tail:N \l__graphics_backend_ext_str
1710       -converted-to.pdf
1711     }
1712     {#1}
1713   }
1714   \cs_new_protected:Npn \__graphics_backend_getbb_eps:nn #1#2
1715   {
1716     \file_compare_timestamp:nNnT {#2} > {#1}
1717     {
1718       \sys_shell_now:n
1719       { repstopdf ~ #2 ~ #1 }
1720     }
1721     \tl_set:Nn \l__graphics_name_tl {#1}
1722     \__graphics_backend_getbb_pdf:n {#1}
1723   }
1724   \cs_new_protected:Npn \__graphics_backend_include_eps:n #1
1725   {
1726     \file_parse_full_name:nNNN {#1}
1727     \l__graphics_backend_dir_str \l__graphics_backend_name_str \l__graphics_backend_ext_str
1728     \exp_args:Nx \__graphics_backend_include_pdf:n
1729     {
1730       \l__graphics_backend_name_str - \str_tail:N \l__graphics_backend_ext_str
1731       -converted-to.pdf
1732     }
1733   }
1734 }

```

(End definition for \\_\_graphics\_backend\_getbb\_eps:n and others.)

```

1735 </luatex | pdftex>

```

### 5.3 dvipdfmx backend

1736 `<*dvipdfmx | xetex>`

`\_graphics_backend_getbb_eps:n`  
`\_graphics_backend_getbb_jpg:n`  
`\_graphics_backend_getbb_pdf:n`  
`\_graphics_backend_getbb_png:n`

Simply use the generic functions: only for dvipdfmx in the extraction cases.

```
1737 \cs_new_eq:NN \_graphics_backend_getbb_eps:n \graphics_read_bb:n
1738 <*dvipdfmx>
1739 \cs_new_protected:Npn \_graphics_backend_getbb_jpg:n #1
1740 {
1741   \int_zero:N \l_graphics_page_int
1742   \tl_clear:N \l_graphics_pagebox_tl
1743   \graphics_extract_bb:n {#1}
1744 }
1745 \cs_new_eq:NN \_graphics_backend_getbb_png:n \_graphics_backend_getbb_jpg:n
1746 \cs_new_protected:Npn \_graphics_backend_getbb_pdf:n #1
1747 {
1748   \tl_clear:N \l_graphics_decodearray_tl
1749   \bool_set_false:N \l_graphics_interpolate_bool
1750   \graphics_extract_bb:n {#1}
1751 }
1752 </dvipdfmx>
```

(End definition for `\_graphics_backend_getbb_eps:n` and others.)

`\g__graphics_track_int`

Used to track the object number associated with each graphic.

```
1753 \int_new:N \g__graphics_track_int
```

(End definition for `\g__graphics_track_int`.)

`\_graphics_backend_include_eps:n`  
`\_graphics_backend_include_jpg:n`  
`\_graphics_backend_include_pdf:n`  
`\_graphics_backend_include_png:n`  
`\_graphics_backend_include_auxi:nn`  
`\_graphics_backend_include_auxii:nnn`  
`\_graphics_backend_include_auxiii:nnn`

The special syntax depends on the file type. There is a difference in how PDF graphics are best handled between dvipdfmx and Xe<sub>La</sub>TeX: for the latter it is better to use the primitive route. The relevant code for that is included later in this file.

```
1754 \cs_new_protected:Npn \_graphics_backend_include_eps:n #1
1755 {
1756   \__kernel_backend_literal:x
1757   {
1758     PSfile = #1 \c_space_tl
1759     llx = \dim_to_decimal_in_bp:n \l_graphics_llx_dim \c_space_tl
1760     lly = \dim_to_decimal_in_bp:n \l_graphics_lly_dim \c_space_tl
1761     urx = \dim_to_decimal_in_bp:n \l_graphics_urx_dim \c_space_tl
1762     ury = \dim_to_decimal_in_bp:n \l_graphics_ury_dim
1763   }
1764 }
1765 \cs_new_protected:Npn \_graphics_backend_include_jpg:n #1
1766 { \_graphics_backend_include_auxi:nn {#1} { image } }
1767 \cs_new_eq:NN \_graphics_backend_include_png:n \_graphics_backend_include_jpg:n
1768 <*dvipdfmx>
1769 \cs_new_protected:Npn \_graphics_backend_include_pdf:n #1
1770 { \_graphics_backend_include_auxi:nn {#1} { epdf } }
1771 </dvipdfmx>
```

Graphic inclusion is set up to use the fact that each image is stored in the PDF as an XObject. This means that we can include repeated images only once and refer to them. To allow that, track the nature of each image: much the same as for the direct PDF mode case.

```

1772 \cs_new_protected:Npn \__graphics_backend_include_auxi:nn #1#2
1773 {
1774   \__graphics_backend_include_auxii:xnn
1775   {
1776     \tl_if_empty:NF \l_graphics_pagebox_tl
1777     { : \l_graphics_pagebox_tl }
1778     \int_compare:nNnT \l_graphics_page_int > 1
1779     { :P \int_use:N \l_graphics_page_int }
1780     \tl_if_empty:NF \l_graphics_decodearray_tl
1781     { :D \l_graphics_decodearray_tl }
1782     \bool_if:NT \l_graphics_interpolate_bool
1783     { :I }
1784   }
1785   {#1} {#2}
1786 }
1787 \cs_new_protected:Npn \__graphics_backend_include_auxii:nnn #1#2#3
1788 {
1789   \int_if_exist:cTF { c__graphics_graphics_ #2#1 _int }
1790   {
1791     \__kernel_backend_literal:x
1792     { pdf:usexobj~@graphic \int_use:c { c__graphics_graphics_ #2#1 _int } }
1793   }
1794   { \__graphics_backend_include_auxiii:nnn {#2} {#1} {#3} }
1795 }
1796 \cs_generate_variant:Nn \__graphics_backend_include_auxii:nnn { x }

```

Inclusion using the specials is relatively straight-forward, but there is one wrinkle. To get the `pagebox` correct for PDF graphics in all cases, it is necessary to provide both that information and the `bbox` argument: odd things happen otherwise!

```

1797 \cs_new_protected:Npn \__graphics_backend_include_auxiii:nnn #1#2#3
1798 {
1799   \int_gincr:N \g__graphics_track_int
1800   \int_const:cn { c__graphics_graphics_ #1#2 _int } { \g__graphics_track_int }
1801   \__kernel_backend_literal:x
1802   {
1803     pdf:#3~
1804     @graphic \int_use:c { c__graphics_graphics_ #1#2 _int } ~
1805     \int_compare:nNnT \l_graphics_page_int > 1
1806     { page ~ \int_use:N \l_graphics_page_int \c_space_tl }
1807     \tl_if_empty:NF \l_graphics_pagebox_tl
1808     {
1809       pagebox ~ \l_graphics_pagebox_tl \c_space_tl
1810       bbox ~
1811       \dim_to_decimal_in_bp:n \l_graphics_llx_dim \c_space_tl
1812       \dim_to_decimal_in_bp:n \l_graphics_lly_dim \c_space_tl
1813       \dim_to_decimal_in_bp:n \l_graphics_urx_dim \c_space_tl
1814       \dim_to_decimal_in_bp:n \l_graphics_ury_dim \c_space_tl
1815     }
1816     (#1)
1817     \bool_lazy_or:nnT
1818     { \l_graphics_interpolate_bool }
1819     { ! \tl_if_empty_p:N \l_graphics_decodearray_tl }
1820     {
1821       <<

```

```

1822         \tl_if_empty:NF \l_graphics_decodearray_tl
1823         { /Decode~[ \l_graphics_decodearray_tl ] }
1824         \bool_if:NT \l_graphics_interpolate_bool
1825         { /Interpolate~true> }
1826     >>
1827 }
1828 }
1829 }

```

(End definition for `\__graphics_backend_include_eps:n` and others.)

```
1830 </dvipdfmx | xetex>
```

## 5.4 X<sub>Y</sub>TeX backend

```
1831 <*xetex>
```

### 5.4.1 Images

For X<sub>Y</sub>TeX, there are two primitives that allow us to obtain the bounding box without needing `extractbb`. The only complexity is passing the various minor variations to a common core process. The X<sub>Y</sub>TeX primitive omits the text box from the page box specification, so there is also some “trimming” to do here.

```

\__graphics_backend_getbb_jpg:n
\__graphics_backend_getbb_pdf:n
\__graphics_backend_getbb_png:n
\__graphics_backend_getbb_auxi:nN
\__graphics_backend_getbb_auxii:nNn
\__graphics_backend_getbb_auxiii:nNnn
\__graphics_backend_getbb_auxiv:nNnn
\__graphics_backend_getbb_auxv:nNnn
\__graphics_backend_getbb_auxvi:nNnn
\__graphics_backend_getbb_pagebox:w

```

```

1832 \cs_new_protected:Npn \__graphics_backend_getbb_jpg:n #1
1833 {
1834     \int_zero:N \l_graphics_page_int
1835     \tl_clear:N \l_graphics_pagebox_tl
1836     \__graphics_backend_getbb_auxi:nN {#1} \tex_XeTeXpicfile:D
1837 }
1838 \cs_new_eq:NN \__graphics_backend_getbb_png:n \__graphics_backend_getbb_jpg:n
1839 \cs_new_protected:Npn \__graphics_backend_getbb_pdf:n #1
1840 {
1841     \tl_clear:N \l_graphics_decodearray_tl
1842     \bool_set_false:N \l_graphics_interpolate_bool
1843     \__graphics_backend_getbb_auxi:nN {#1} \tex_XeTeXpdffile:D
1844 }
1845 \cs_new_protected:Npn \__graphics_backend_getbb_auxi:nN #1#2
1846 {
1847     \int_compare:nNnTF \l_graphics_page_int > 1
1848     { \__graphics_backend_getbb_auxii:VnN \l_graphics_page_int {#1} #2 }
1849     { \__graphics_backend_getbb_auxiii:nNnn {#1} #2 { :P 1 } { page 1 } }
1850 }
1851 \cs_new_protected:Npn \__graphics_backend_getbb_auxii:nN #1#2#3
1852 { \__graphics_backend_getbb_auxiii:nNnn {#2} #3 { :P #1 } { page #1 } }
1853 \cs_generate_variant:Nn \__graphics_backend_getbb_auxii:nN { V }
1854 \cs_new_protected:Npn \__graphics_backend_getbb_auxiii:nNnn #1#2#3#4
1855 {
1856     \tl_if_empty:NTF \l_graphics_pagebox_tl
1857     { \__graphics_backend_getbb_auxiv:VnNnn \l_graphics_pagebox_tl }
1858     { \__graphics_backend_getbb_auxv:nNnn }
1859     {#1} #2 {#3} {#4}
1860 }
1861 \cs_new_protected:Npn \__graphics_backend_getbb_auxiv:nNnn #1#2#3#4#5
1862 {
1863     \use:x

```



```

1864     {
1865         \__graphics_backend_getbb_auxv:nNnn {#2} #3 { : #1 #4 }
1866         { #5 ~ \__graphics_backend_getbb_pagebox:w #1 }
1867     }
1868 }
1869 \cs_generate_variant:Nn \__graphics_backend_getbb_auxiv:nnNnn { V }
1870 \cs_new_protected:Npn \__graphics_backend_getbb_auxv:nNnn #1#2#3#4
1871 {
1872     \graphics_bb_restore:nF {#1#3}
1873     { \__graphics_backend_getbb_auxvi:nNnn {#1} #2 {#3} {#4} }
1874 }
1875 \cs_new_protected:Npn \__graphics_backend_getbb_auxvi:nNnn #1#2#3#4
1876 {
1877     \hbox_set:Nn \l__graphics_internal_box { #2 #1 ~ #4 }
1878     \dim_set:Nn \l_graphics_urx_dim { \box_wd:N \l__graphics_internal_box }
1879     \dim_set:Nn \l_graphics_ury_dim { \box_ht:N \l__graphics_internal_box }
1880     \graphics_bb_save:n {#1#3}
1881 }
1882 \cs_new:Npn \__graphics_backend_getbb_pagebox:w #1 box {#1}

```

(End definition for \\_\_graphics\_backend\_getbb\_jpg:n and others.)

\\_\_graphics\_backend\_include\_pdf:n  
\\_\_graphics\_backend\_include\_bitmap\_quote:w

For PDF graphics, properly supporting the pagebox concept in X<sub>Y</sub>TeX is best done using the \tex\_XeTeXpdf\_file:D primitive. The syntax here is the same as for the graphic measurement part, although we know at this stage that there must be some valid setting for \l\_graphics\_pagebox\_tl.

```

1883 \cs_new_protected:Npn \__graphics_backend_include_pdf:n #1
1884 {
1885     \tex_XeTeXpdf_file:D
1886     \__graphics_backend_include_pdf_quote:w #1 "#1" \s__graphics_stop \c_space_tl
1887     \int_compare:nNnT \l_graphics_page_int > 0
1888         { page ~ \int_use:N \l_graphics_page_int \c_space_tl }
1889     \exp_after:wN \__graphics_backend_getbb_pagebox:w \l_graphics_pagebox_tl
1890 }
1891 \cs_new:Npn \__graphics_backend_include_pdf_quote:w #1 " #2 " #3 \s__graphics_stop
1892 { " #2 " }

```

(End definition for \\_\_graphics\_backend\_include\_pdf:n and \\_\_graphics\_backend\_include\_bitmap\_quote:w.)

```
1893 </xetex>
```

## 5.5 dvisvgm backend

```
1894 <*dvisvgm>
```

\\_\_graphics\_backend\_getbb\_eps:n

Simply use the generic function.

```
1895 \cs_new_eq:NN \__graphics_backend_getbb_eps:n \graphics_read_bb:n
```

(End definition for \\_\_graphics\_backend\_getbb\_eps:n.)

\\_\_graphics\_backend\_getbb\_png:n

These can be included by extracting the bounding box data.

\\_\_graphics\_backend\_getbb\_jpg:n

```

1896 \cs_new_protected:Npn \__graphics_backend_getbb_jpg:n #1
1897 {
1898     \int_zero:N \l_graphics_page_int

```

```

1899 \tl_clear:N \l_graphics_pagebox_tl
1900 \graphics_extract_bb:n {#1}
1901 }
1902 \cs_new_eq:NN \__graphics_backend_getbb_png:n \__graphics_backend_getbb_jpg:n
(End definition for \__graphics_backend_getbb_png:n and \__graphics_backend_getbb_jpg:n.)

```

\\_\_graphics\_backend\_getbb\_pdf:n Same as for dvipdfmx: use the generic function

```

1903 \cs_new_protected:Npn \__graphics_backend_getbb_pdf:n #1
1904 {
1905   \tl_clear:N \l_graphics_decodearray_tl
1906   \bool_set_false:N \l_graphics_interpolate_bool
1907   \graphics_extract_bb:n {#1}
1908 }
(End definition for \__graphics_backend_getbb_pdf:n.)

```

\\_\_graphics\_backend\_include\_eps:n The special syntax is relatively clear here: remember we need PostScript sizes here. (This is the same as the dvips code.)

```

\__graphics_backend_include_pdf:n
\__graphics_backend_include:nn
1909 \cs_new_protected:Npn \__graphics_backend_include_eps:n #1
1910 { __graphics_backend_include:nn { PSfile } {#1} }
1911 \cs_new_protected:Npn \__graphics_backend_include_pdf:n #1
1912 { __graphics_backend_include:nn { pdffile } {#1} }
1913 \cs_new_protected:Npn \__graphics_backend_include:nn #1#2
1914 {
1915   \__kernel_backend_literal:x
1916   {
1917     #1 = #2 \c_space_tl
1918     llx = \dim_to_decimal_in_bp:n \l_graphics_llx_dim \c_space_tl
1919     lly = \dim_to_decimal_in_bp:n \l_graphics_lly_dim \c_space_tl
1920     urx = \dim_to_decimal_in_bp:n \l_graphics_urx_dim \c_space_tl
1921     ury = \dim_to_decimal_in_bp:n \l_graphics_ury_dim
1922   }
1923 }
(End definition for \__graphics_backend_include_eps:n, \__graphics_backend_include_pdf:n, and \__graphics_backend_include:nn.)

```

\\_\_graphics\_backend\_include\_png:n The backend here has built-in support for basic graphic inclusion (see dvisvgm.def for a more complex approach, needed if clipping, *etc.*, is covered at the graphic backend level).

\\_\_graphics\_backend\_include\_jpg:n The only issue is that #1 must be quote-corrected. The dvisvgm:img operation quotes the file name, but if it is already quoted (contains spaces) then we have an issue: we simply strip off any quotes as a result.

\\_\_graphics\_backend\_include\_bitmap\_quote:w

```

1924 \cs_new_protected:Npn \__graphics_backend_include_png:n #1
1925 {
1926   \__kernel_backend_literal:x
1927   {
1928     dvisvgm:img~
1929     \dim_to_decimal:n { \l_graphics_ury_dim } ~
1930     \dim_to_decimal:n { \l_graphics_ury_dim } ~
1931     \__graphics_backend_include_bitmap_quote:w #1 " #1 " \s__graphics_stop
1932   }
1933 }
1934 \cs_new_eq:NN \__graphics_backend_include_jpg:n \__graphics_backend_include_png:n
1935 \cs_new:Npn \__graphics_backend_include_bitmap_quote:w #1 " #2 " #3 \s__graphics_stop
1936 { " #2 " }

```

(End definition for `\_graphics_backend_include_png:n`, `\_graphics_backend_include_jpg:n`, and `\_graphics_backend_include_bitmap_quote:w`.)

1937 `\dvism`

1938 `\package`

## 6 l3backend-pdf Implementation

1939 `*package`

1940 `@=pdf`

Setting up PDF resources is a complex area with only limited documentation in the engine manuals. The following code builds heavily on existing ideas from `hyperref` work by Sebastian Rahtz and Heiko Oberdiek, and significant contributions by Alexander Grahn, in addition to the specific code referenced a various points.

### 6.1 Shared code

A very small number of items that belong at the backend level but which are common to all backends.

`\l__pdf_internal_box`

1941 `\box_new:N \l__pdf_internal_box`

(End definition for `\l__pdf_internal_box`.)

### 6.2 dvips backend

1942 `*dvips`

`\_pdf_backend_pdfmark:n`

`\_pdf_backend_pdfmark:x`

Used often enough it should be a separate function.

1943 `\cs_new_protected:Npn \_pdf_backend_pdfmark:n #1`

1944 `{ \_kernel_backend_postscript:n { mark #1 ~ pdfmark } }`

1945 `\cs_generate_variant:Nn \_pdf_backend_pdfmark:n { x }`

(End definition for `\_pdf_backend_pdfmark:n`.)

#### 6.2.1 Catalogue entries

`\_pdf_backend_catalog_gput:nn`

`\_pdf_backend_info_gput:nn`

1946 `\cs_new_protected:Npn \_pdf_backend_catalog_gput:nn #1#2`

1947 `{ \_pdf_backend_pdfmark:n { { Catalog } << /#1 ~ #2 >> /PUT } }`

1948 `\cs_new_protected:Npn \_pdf_backend_info_gput:nn #1#2`

1949 `{ \_pdf_backend_pdfmark:n { /#1 ~ #2 /DOCINFO } }`

(End definition for `\_pdf_backend_catalog_gput:nn` and `\_pdf_backend_info_gput:nn`.)

## 6.2.2 Objects

`\g__pdf_backend_object_int`  
`\g__pdf_backend_object_prop`

For tracking objects to allow finalisation.

```
1950 \int_new:N \g__pdf_backend_object_int
1951 \prop_new:N \g__pdf_backend_object_prop
```

(End definition for `\g__pdf_backend_object_int` and `\g__pdf_backend_object_prop`.)

`\__pdf_backend_object_new:nn`  
`\__pdf_backend_object_ref:n`

Tracking objects is similar to dvipdfmx.

```
1952 \cs_new_protected:Npn \__pdf_backend_object_new:nn #1#2
1953 {
1954   \int_gincr:N \g__pdf_backend_object_int
1955   \int_const:cn
1956     { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
1957     { \g__pdf_backend_object_int }
1958   \prop_gput:Nnn \g__pdf_backend_object_prop {#1} {#2}
1959 }
1960 \cs_new:Npn \__pdf_backend_object_ref:n #1
1961 { { pdf.obj \int_use:c { c__pdf_backend_object_ \tl_to_str:n {#1} _int } } }
```

(End definition for `\__pdf_backend_object_new:nn` and `\__pdf_backend_object_ref:n`.)

`\__pdf_backend_object_write:nn`  
`\__pdf_backend_object_write:nx`  
`\__pdf_backend_object_write_array:nn`  
`\__pdf_backend_object_write_dict:nn`  
`\__pdf_backend_object_write_fstream:nn`  
`\__pdf_backend_object_write_stream:nn`  
`\__pdf_backend_object_write_stream:nmn`

This is where we choose the actual type: some work to get things right.

```
1962 \cs_new_protected:Npn \__pdf_backend_object_write:nn #1#2
1963 {
1964   \__pdf_backend_pdfmark:x
1965   {
1966     /_objdef ~ \__pdf_backend_object_ref:n {#1}
1967     /type
1968     \str_case_e:nn
1969       { \prop_item:Nn \g__pdf_backend_object_prop {#1} }
1970       {
1971         { array } { /array }
1972         { dict } { /dict }
1973         { fstream } { /stream }
1974         { stream } { /stream }
1975       }
1976     /OBJ
1977   }
1978   \use:c
1979     { __pdf_backend_object_write_ \prop_item:Nn \g__pdf_backend_object_prop {#1} :nn }
1980     { \__pdf_backend_object_ref:n {#1} } {#2}
1981 }
1982 \cs_generate_variant:Nn \__pdf_backend_object_write:nn { nx }
1983 \cs_new_protected:Npn \__pdf_backend_object_write_array:nn #1#2
1984 {
1985   \__pdf_backend_pdfmark:x
1986     { #1 ~0~ [ ~ \exp_not:n {#2} ~ ] ~ /PUTINTERVAL }
1987 }
1988 \cs_new_protected:Npn \__pdf_backend_object_write_dict:nn #1#2
1989 {
1990   \__pdf_backend_pdfmark:x
1991     { #1 << \exp_not:n {#2} >> /PUT }
1992 }
1993 \cs_new_protected:Npn \__pdf_backend_object_write_fstream:nn #1#2
```

```

1994 {
1995   \exp_args:Nx
1996   \__pdf_backend_object_write_fstream:nnn {#1} #2
1997 }
1998 \cs_new_protected:Npn \__pdf_backend_object_write_fstream:nnn #1#2#3
1999 {
2000   \__kernel_backend_postscript:n
2001   {
2002     SDict ~ begin ~
2003     mark ~ #1 ~ << #2 >> /PUT ~ pdfmark ~
2004     mark ~ #1 ~ ( #3 )~ ( r )~ file ~ /PUT ~ pdfmark ~
2005     end
2006   }
2007 }
2008 \cs_new_protected:Npn \__pdf_backend_object_write_stream:nn #1#2
2009 {
2010   \exp_args:Nx
2011   \__pdf_backend_object_write_stream:nnn {#1} #2
2012 }
2013 \cs_new_protected:Npn \__pdf_backend_object_write_stream:nnn #1#2#3
2014 {
2015   \__kernel_backend_postscript:n
2016   {
2017     mark ~ #1 ~ ( #3 ) /PUT ~ pdfmark ~
2018     mark ~ #1 ~ << #2 >> /PUT ~ pdfmark
2019   }
2020 }

```

(End definition for \\_\_pdf\_backend\_object\_write:nn and others.)

\\_\_pdf\_backend\_object\_now:nn  
\\_\_pdf\_backend\_object\_now:nx

No anonymous objects, so things are done manually.

```

2021 \cs_new_protected:Npn \__pdf_backend_object_now:nn #1#2
2022 {
2023   \int_gincr:N \g__pdf_backend_object_int
2024   \__pdf_backend_pdfmark:x
2025   {
2026     /objdef ~ { pdf.obj \int_use:N \g__pdf_backend_object_int }
2027     /type
2028     \str_case:nn
2029       {#1}
2030       {
2031         { array } { /array }
2032         { dict } { /dict }
2033         { fstream } { /stream }
2034         { stream } { /stream }
2035       }
2036     /OBJ
2037   }
2038   \exp_args:Nnx \use:c { __pdf_backend_object_write_ #1 :nn }
2039   { { pdf.obj \int_use:N \g__pdf_backend_object_int } } {#2}
2040 }
2041 \cs_generate_variant:Nn \__pdf_backend_object_now:nn { nx }

```

(End definition for \\_\_pdf\_backend\_object\_now:nn.)

`\__pdf_backend_object_last:` Much like the annotation version.

```

2042 \cs_new:Npn \__pdf_backend_object_last:
2043 { { pdf.obj \int_use:N \g__pdf_backend_object_int } }

(End definition for \__pdf_backend_object_last:.)

```

`\__pdf_backend_pageobject_ref:n` Page references are easy in dvips.

```

2044 \cs_new:Npn \__pdf_backend_pageobject_ref:n #1
2045 { { Page #1 } }

(End definition for \__pdf_backend_pageobject_ref:n.)

```

### 6.2.3 Annotations

In dvips, annotations have to be constructed manually. As such, we need the object code above for some definitions.

`\l__pdf_backend_content_box` The content of an annotation.

```

2046 \box_new:N \l__pdf_backend_content_box

(End definition for \l__pdf_backend_content_box.)

```

`\l__pdf_backend_model_box` For creating model sizing for links.

```

2047 \box_new:N \l__pdf_backend_model_box

(End definition for \l__pdf_backend_model_box.)

```

`\g__pdf_backend_annotation_int` Needed as objects which are not annotations could be created.

```

2048 \int_new:N \g__pdf_backend_annotation_int

(End definition for \g__pdf_backend_annotation_int.)

```

`\__pdf_backend_annotation:nnnn` Annotations are objects, but we track them separately. Notably, they are not in the object data lists. Here, to get the co-ordinates of the annotation, we need to have the data collected at the PostScript level. That requires a bit of box trickery (effectively a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> picture of zero size). Once the data is collected, use it to set up the annotation border.

```

2049 \cs_new_protected:Npn \__pdf_backend_annotation:nnnn #1#2#3#4
2050 {
2051   \exp_args:Nf \__pdf_backend_annotation_aux:nnnn
2052     { \dim_eval:n {#1} } {#2} {#3} {#4}
2053 }
2054 \cs_new_protected:Npn \__pdf_backend_annotation_aux:nnnn #1#2#3#4
2055 {
2056   \box_move_down:nn {#3}
2057   { \hbox:n { \__kernel_backend_postscript:n { pdf.save.ll } } }
2058   \box_move_up:nn {#2}
2059   {
2060     \hbox:n
2061     {
2062       \tex_kern:D #1 \scan_stop:
2063       \__kernel_backend_postscript:n { pdf.save.ur }
2064       \tex_kern:D -#1 \scan_stop:
2065     }

```

```

2066     }
2067     \int_gincr:N \g__pdf_backend_object_int
2068     \int_gset_eq:NN \g__pdf_backend_annotation_int \g__pdf_backend_object_int
2069     \__pdf_backend_pdfmark:x
2070     {
2071         /_objdef { pdf.obj \int_use:N \g__pdf_backend_object_int }
2072         pdf.rect
2073         #4 ~
2074         /ANN
2075     }
2076 }

```

(End definition for \\_\_pdf\_backend\_annotation:nnnn.)

\\_\_pdf\_backend\_annotation\_last: Provide the last annotation we created: could get tricky of course if other packages are loaded.

```

2077 \cs_new:Npn \__pdf_backend_annotation_last:
2078 { { pdf.obj \int_use:N \g__pdf_backend_annotation_int } }

```

(End definition for \\_\_pdf\_backend\_annotation\_last:.)

\g\_\_pdf\_backend\_link\_int To track annotations which are links.

```

2079 \int_new:N \g__pdf_backend_link_int

```

(End definition for \g\_\_pdf\_backend\_link\_int.)

\g\_\_pdf\_backend\_link\_dict\_tl To pass information to the end-of-link function.

```

2080 \tl_new:N \g__pdf_backend_link_dict_tl

```

(End definition for \g\_\_pdf\_backend\_link\_dict\_tl.)

\g\_\_pdf\_backend\_link\_sf\_int Needed to save/restore space factor, which is needed to deal with the face we need a box.

```

2081 \int_new:N \g__pdf_backend_link_sf_int

```

(End definition for \g\_\_pdf\_backend\_link\_sf\_int.)

\g\_\_pdf\_backend\_link\_math\_bool Needed to save/restore math mode.

```

2082 \bool_new:N \g__pdf_backend_link_math_bool

```

(End definition for \g\_\_pdf\_backend\_link\_math\_bool.)

\g\_\_pdf\_backend\_link\_bool Track link formation: we cannot nest at all.

```

2083 \bool_new:N \g__pdf_backend_link_bool

```

(End definition for \g\_\_pdf\_backend\_link\_bool.)

\l\_\_pdf\_breaklink\_pdfmark\_tl Swappable content for link breaking.

```

2084 \tl_new:N \l__pdf_breaklink_pdfmark_tl
2085 \tl_set:Nn \l__pdf_breaklink_pdfmark_tl { pdfmark }

```

(End definition for \l\_\_pdf\_breaklink\_pdfmark\_tl.)

\\_\_pdf\_breaklink\_postscript:n To allow dropping material unless link breaking is active.

```

2086 \cs_new_protected:Npn \__pdf_breaklink_postscript:n #1 { }

```

(End definition for \\_\_pdf\_breaklink\_postscript:n.)

`\_pdf_breaklink_usebox:N` Swappable box unpacking or use.

2087 `\cs_new_eq:NN \_pdf_breaklink_usebox:N \box_use:N`

(End definition for `\_pdf_breaklink_usebox:N`.)

Links are crated like annotations but with dedicated code to allow for adjusting the size of the rectangle. In contrast to `hyperref`, we grab the link content as a box which can then unbox: this allows the same interface as for `pdfTeX`.

Taking the idea of `evenboxes` from `hypdvips`, we implement a minimum box height and depth for link placement. This means that “underlining” with a hyperlink will generally give an even appearance. However, to ensure that the full content is always above the link border, we do not allow this to be negative (contrast `hypdvips` approach). The result should be similar to `pdfTeX` in the vast majority of foreseeable cases.

The object number for a link is saved separately from the rest of the dictionary as this allows us to insert it just once, at either an unbroken link or only in the first line of a broken one. That makes the code clearer but also avoids a low-level PostScript error with the code as taken from `hypdvips`.

Getting the outer dimensions of the text area may be better using a two-pass approach and `\tex_savepos:D`. That plus format mode are still to re-examine.

```

pdf.linkdp.pad
pdf.linkht.pad
pdf.llx
pdf.lly
pdf.ury
pdf.link.dict
pdf.outerbox
pdf.baselineskip
2088 \cs_new_protected:Npn \_pdf_backend_link_begin_goto:nnw #1#2
2089 { \_pdf_backend_link_begin:nw { #1 /Subtype /Link /A << /S /GoTo /D ( #2 ) >> } }
2090 \cs_new_protected:Npn \_pdf_backend_link_begin_user:nnw #1#2
2091 { \_pdf_backend_link_begin:nw {#1#2} }
2092 \cs_new_protected:Npn \_pdf_backend_link_begin:nw #1
2093 {
2094   \bool_if:NF \g__pdf_backend_link_bool
2095   { \_pdf_backend_link_begin_aux:nw {#1} }
2096 }
2097 \cs_new_protected:Npn \_pdf_backend_link_begin_aux:nw #1
2098 {
2099   \bool_gset_true:N \g__pdf_backend_link_bool
2100   \_kernel_backend_postscript:n
2101   { /pdf.link.dict ( #1 ) def }
2102   \tl_gset:Nn \g__pdf_backend_link_dict_tl {#1}
2103   \_pdf_backend_link_sf_save:
2104   \mode_if_math:TF
2105   { \bool_gset_true:N \g__pdf_backend_link_math_bool }
2106   { \bool_gset_false:N \g__pdf_backend_link_math_bool }
2107   \hbox_set:Nw \l__pdf_backend_content_box
2108   \_pdf_backend_link_sf_restore:
2109   \bool_if:NT \g__pdf_backend_link_math_bool
2110   { \c_math_toggle_token }
2111 }
2112 \cs_new_protected:Npn \_pdf_backend_link_end:
2113 {
2114   \bool_if:NT \g__pdf_backend_link_bool
2115   { \_pdf_backend_link_end_aux: }
2116 }
2117 \cs_new_protected:Npn \_pdf_backend_link_end_aux:
2118 {
2119   \bool_if:NT \g__pdf_backend_link_math_bool
2120   { \c_math_toggle_token }
2121   \_pdf_backend_link_sf_save:

```



```

2122 \hbox_set_end:
2123 \__pdf_backend_link_minima:
2124 \hbox_set:Nn \l__pdf_backend_model_box { Gg }
2125 \exp_args:Nx \__pdf_backend_link_outerbox:n
2126 {
2127   \int_if_odd:nTF { \value { page } }
2128     { \oddsidemargin }
2129     { \evensidemargin }
2130 }
2131 \box_move_down:nn { \box_dp:N \l__pdf_backend_content_box }
2132 { \hbox:n { \__kernel_backend_postscript:n { pdf.save.linkll } } }
2133 \__pdf_breaklink_postscript:n { pdf.bordertracking.begin }
2134 \__pdf_breaklink_usebox:N \l__pdf_backend_content_box
2135 \__pdf_breaklink_postscript:n { pdf.bordertracking.end }
2136 \box_move_up:nn { \box_ht:N \l__pdf_backend_content_box }
2137 {
2138   \hbox:n
2139   { \__kernel_backend_postscript:n { pdf.save.linkur } }
2140 }
2141 \int_gincr:N \g__pdf_backend_object_int
2142 \int_gset_eq:NN \g__pdf_backend_link_int \g__pdf_backend_object_int
2143 \__kernel_backend_postscript:x
2144 {
2145   mark
2146   /objdef { pdf.obj \int_use:N \g__pdf_backend_link_int }
2147   \g__pdf_backend_link_dict_tl \c_space_tl
2148   pdf.rect
2149   /ANN ~ \l__pdf_breaklink_pdfmark_tl
2150 }
2151 \__pdf_backend_link_sf_restore:
2152 \bool_gset_false:N \g__pdf_backend_link_bool
2153 }
2154 \cs_new_protected:Npn \__pdf_backend_link_minima:
2155 {
2156   \hbox_set:Nn \l__pdf_backend_model_box { Gg }
2157   \__kernel_backend_postscript:x
2158   {
2159     /pdf.linkdp.pad ~
2160     \dim_to_decimal:n
2161     {
2162       \dim_max:nn
2163       {
2164         \box_dp:N \l__pdf_backend_model_box
2165         - \box_dp:N \l__pdf_backend_content_box
2166       }
2167       { Opt }
2168     } ~
2169     pdf.pt.dvi ~ def
2170     /pdf.linkht.pad ~
2171     \dim_to_decimal:n
2172     {
2173       \dim_max:nn
2174       {
2175         \box_ht:N \l__pdf_backend_model_box

```

```

2176         - \box_ht:N \l__pdf_backend_content_box
2177     }
2178     { Opt }
2179 } ~
2180 pdf.pt.dvi ~ def
2181 }
2182 }
2183 \cs_new_protected:Npn \__pdf_backend_link_outerbox:n #1
2184 {
2185     \__kernel_backend_postscript:x
2186     {
2187         /pdf.outerbox
2188         [
2189             \dim_to_decimal:n {#1} ~
2190             \dim_to_decimal:n { -\box_dp:N \l__pdf_backend_model_box } ~
2191             \dim_to_decimal:n { #1 + \textwidth } ~
2192             \dim_to_decimal:n { \box_ht:N \l__pdf_backend_model_box }
2193         ]
2194         [ exch { pdf.pt.dvi } forall ] def
2195         /pdf.baselineskip ~
2196         \dim_to_decimal:n { \tex_baselineskip:D } ~ dup ~ 0 ~ gt
2197         { pdf.pt.dvi ~ def }
2198         { pop ~ pop }
2199     ifelse
2200 }
2201 }
2202 \cs_new_protected:Npn \__pdf_backend_link_sf_save:
2203 {
2204     \int_gset:Nn \g__pdf_backend_link_sf_int
2205     {
2206         \mode_if_horizontal:TF
2207         { \tex_spacefactor:D }
2208         { 0 }
2209     }
2210 }
2211 \cs_new_protected:Npn \__pdf_backend_link_sf_restore:
2212 {
2213     \mode_if_horizontal:T
2214     {
2215         \int_compare:nNnT \g__pdf_backend_link_sf_int > { 0 }
2216         { \int_set_eq:NN \tex_spacefactor:D \g__pdf_backend_link_sf_int }
2217     }
2218 }

```

(End definition for `\__pdf_backend_link_begin_goto:nw` and others. These functions are documented on page ??.)

`\@makecol@hook` Hooks to allow link breaking: something will be needed in format mode at some stage. At present this code is disabled as there is an open question about the name of the hook: to be resolved at the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> end.

```

2219 \use_none:n
2220 {
2221     \cs_if_exist:NT \@makecol@hook
2222     {

```

```

2223     \tl_put_right:Nn \@makecol@hook
2224     {
2225         \box_if_empty:NF \@cclv
2226         {
2227             \vbox_set:Nn \@cclv
2228             {
2229                 \__kernel_backend_postscript:n
2230                 {
2231                     pdf.globaldict /pdf.brokenlink.rect ~ known
2232                     { pdf.bordertracking.continue }
2233                     if
2234                     }
2235                     \vbox_unpack_drop:N \@cclv
2236                     \__kernel_backend_postscript:n
2237                     { pdf.bordertracking.endpage }
2238                 }
2239             }
2240         }
2241     \tl_set:Nn \l__pdf_breaklink_pdfmark_tl { pdf.pdfmark }
2242     \cs_set_eq:NN \__pdf_breaklink_postscript:n \__kernel_backend_postscript:n
2243     \cs_set_eq:NN \__pdf_breaklink_usebox:N \hbox_unpack:N
2244 }
2245 }

```

(End definition for \@makecol@hook. This function is documented on page ??.)

\\_\_pdf\_backend\_link\_last: The same as annotations, but with a custom integer.

```

2246 \cs_new:Npn \__pdf_backend_link_last:
2247 { { pdf.obj \int_use:N \g__pdf_backend_link_int } }

```

(End definition for \\_\_pdf\_backend\_link\_last:.)

\\_\_pdf\_backend\_link\_margin:n Convert to big points and pass to PostScript.

```

2248 \cs_new_protected:Npn \__pdf_backend_link_margin:n #1
2249 {
2250     \__kernel_backend_postscript:x
2251     {
2252         /pdf.linkmargin { \dim_to_decimal:n {#1} ~ pdf.pt.dvi } def
2253     }
2254 }

```

(End definition for \\_\_pdf\_backend\_link\_margin:n.)

\\_\_pdf\_backend\_destination:nn Here, we need to turn the zoom into a scale. We also need to know where the current anchor point actually is: worked out in PostScript. For the rectangle version, we have a bit more PostScript: we need two points. fitr without rule spec doesn't work, so it falls back to /Fit here.

\\_\_pdf\_backend\_destination\_box:nn

```

2255 \cs_new_protected:Npn \__pdf_backend_destination:nn #1#2
2256 {
2257     \__kernel_backend_postscript:n { pdf.dest.anchor }
2258     \__pdf_backend_pdfmark:x
2259     {
2260         /View
2261         [

```

```

2262 \str_case:nnF {#2}
2263 {
2264   { xyz } { /XYZ ~ pdf.dest.point ~ null }
2265   { fit } { /Fit }
2266   { fitb } { /FitB }
2267   { fitbh } { /FitBH ~ pdf.dest.y }
2268   { fitbv } { /FitBV ~ pdf.dest.x }
2269   { fith } { /FitH ~ pdf.dest.y }
2270   { fitv } { /FitV ~ pdf.dest.x }
2271   { fitr } { /Fit }
2272 }
2273 {
2274   /XYZ ~ pdf.dest.point ~ \fp_eval:n { (#2) / 100 }
2275 }
2276 ]
2277 /Dest ( \exp_not:n {#1} ) cvn
2278 /DEST
2279 }
2280 }
2281 \cs_new_protected:Npn \__pdf_backend_destination_box:nn #1#2
2282 {
2283   \group_begin:
2284   \hbox_set:Nn \l__pdf_internal_box {#2}
2285   \box_move_down:nn
2286   { \box_dp:N \l__pdf_internal_box }
2287   { \hbox:n { \__kernel_backend_postscript:n { pdf.save.ll } } }
2288   \box_use:N \l__pdf_internal_box
2289   \box_move_up:nn
2290   { \box_ht:N \l__pdf_internal_box }
2291   { \hbox:n { \__kernel_backend_postscript:n { pdf.save.ur } } }
2292   \__pdf_backend_pdfmark:n
2293   {
2294     /View
2295     [
2296       /FitR ~
2297       pdf.llx ~ pdf.lly ~ pdf.dest2device ~
2298       pdf.urx ~ pdf.ury ~ pdf.dest2device
2299     ]
2300     /Dest ( #1 ) cvn
2301     /DEST
2302   }
2303   \group_end:
2304 }

```

(End definition for \\_\_pdf\_backend\_destination:nn and \\_\_pdf\_backend\_destination\_box:nn.)

## 6.2.4 Structure

\\_\_pdf\_backend\_compresslevel:n Doable for the usual ps2pdf method.

```

\__pdf_backend_compress_objects:n
2305 \cs_new_protected:Npn \__pdf_backend_compresslevel:n #1
2306 {
2307   \int_compare:nNnT {#1} = 0
2308   {
2309     \__kernel_backend_literal_postscript:n

```

```

2310         {
2311             /setdistillerparams ~ where
2312             { pop << /CompressPages ~ false >> setdistillerparams }
2313             if
2314         }
2315     }
2316 }
2317 \cs_new_protected:Npn \__pdf_backend_compress_objects:n #1
2318 {
2319     \bool_if:nF {#1}
2320     {
2321         \__kernel_backend_literal_postscript:n
2322         {
2323             /setdistillerparams ~ where
2324             { pop << /CompressStreams ~ false >> setdistillerparams }
2325             if
2326         }
2327     }
2328 }

```

(End definition for \\_\_pdf\_backend\_compresslevel:n and \\_\_pdf\_backend\_compress\_objects:n.)

\\_\_pdf\_backend\_version\_major\_gset:n  
\\_\_pdf\_backend\_version\_minor\_gset:n

Data not available!

```

2329 \cs_new_protected:Npn \__pdf_backend_version_major_gset:n #1 { }
2330 \cs_new_protected:Npn \__pdf_backend_version_minor_gset:n #1 { }

```

(End definition for \\_\_pdf\_backend\_version\_major\_gset:n and \\_\_pdf\_backend\_version\_minor\_gset:n.)

\\_\_pdf\_backend\_version\_major:  
\\_\_pdf\_backend\_version\_minor:

Data not available!

```

2331 \cs_new:Npn \__pdf_backend_version_major: { -1 }
2332 \cs_new:Npn \__pdf_backend_version_minor: { -1 }

```

(End definition for \\_\_pdf\_backend\_version\_major: and \\_\_pdf\_backend\_version\_minor:.)

### 6.2.5 Marked content

\\_\_pdf\_backend\_bdc:nn  
\\_\_pdf\_backend\_emc:

Simple wrappers.

```

2333 \cs_new_protected:Npn \__pdf_backend_bdc:nn #1#2
2334 { \__pdf_backend_pdfmark:n { /#1 ~ #2 /BDC } }
2335 \cs_new_protected:Npn \__pdf_backend_emc:
2336 { \__pdf_backend_pdfmark:n { /EMC } }

```

(End definition for \\_\_pdf\_backend\_bdc:nn and \\_\_pdf\_backend\_emc:.)

2337 </dvips>

## 6.3 LuaTeX and pdfTeX backend

2338 <\*luatex | pdftex>

### 6.3.1 Annotations

\\_\_pdf\_backend\_annotation:nnnn

Simply pass the raw data through, just dealing with evaluation of dimensions.

```

2339 \cs_new_protected:Npn \__pdf_backend_annotation:nnnn #1#2#3#4
2340 {
2341     <*luatex>

```

```

2342     \tex_pdfextension:D annot ~
2343 </luatex>
2344 <*pdfTeX>
2345     \tex_pdfannot:D
2346 </pdfTeX>
2347     width ~ \dim_eval:n {#1} ~
2348     height ~ \dim_eval:n {#2} ~
2349     depth ~ \dim_eval:n {#3} ~
2350     {#4}
2351 }

```

(End definition for \\_pdf\_backend\_annotation:nnnn.)

\\_pdf\_backend\_annotation\_last: A tiny amount of extra data gets added here; we use x-type expansion to get the space in the right place and form. The “extra” space in the LuaTeX version is *required* as it is consumed in finding the end of the keyword.

```

2352 \cs_new:Npx \_pdf_backend_annotation_last:
2353 {
2354     \exp_not:N \int_value:w
2355 <*luatex>
2356     \exp_not:N \tex_pdffeedback:D lastannot ~
2357 </luatex>
2358 <*pdfTeX>
2359     \exp_not:N \tex_pdflastannot:D
2360 </pdfTeX>
2361     \c_space_tl 0 ~ R
2362 }

```

(End definition for \\_pdf\_backend\_annotation\_last:.)

\\_pdf\_backend\_link\_begin\_goto:nnw  
\\_pdf\_backend\_link\_begin\_user:nnw  
\\_pdf\_backend\_link\_begin:nnnw  
\\_pdf\_backend\_link\_end:

Links are all created using the same internals.

```

2363 \cs_new_protected:Npn \_pdf_backend_link_begin_goto:nnw #1#2
2364 { \_pdf_backend_link_begin:nnnw {#1} { goto~name } {#2} }
2365 \cs_new_protected:Npn \_pdf_backend_link_begin_user:nnw #1#2
2366 { \_pdf_backend_link_begin:nnnw {#1} { user } {#2} }
2367 \cs_new_protected:Npn \_pdf_backend_link_begin:nnnw #1#2#3
2368 {
2369 <*luatex>
2370     \tex_pdfextension:D startlink ~
2371 </luatex>
2372 <*pdfTeX>
2373     \tex_pdfstartlink:D
2374 </pdfTeX>
2375     attr {#1}
2376     #2 {#3}
2377 }
2378 \cs_new_protected:Npn \_pdf_backend_link_end:
2379 {
2380 <*luatex>
2381     \tex_pdfextension:D endlink \scan_stop:
2382 </luatex>
2383 <*pdfTeX>
2384     \tex_pdfendlink:D
2385 </pdfTeX>
2386 }

```

(End definition for `\_pdf_backend_link_begin_goto:nw` and others.)

`\_pdf_backend_link_last:` Formatted for direct use.

```

2387 \cs_new:Npx \_pdf_backend_link_last:
2388 {
2389   \exp_not:N \int_value:w
2390   <*luatex>
2391   \exp_not:N \tex_pdffeedback:D lastlink ~
2392   </luatex>
2393   <*pdftex>
2394   \exp_not:N \tex_pdflastlink:D
2395   </pdftex>
2396   \c_space_tl 0 ~ R
2397 }

```

(End definition for `\_pdf_backend_link_last:.`)

`\_pdf_backend_link_margin:n` A simple task: pass the data to the primitive.

```

2398 \cs_new_protected:Npn \_pdf_backend_link_margin:n #1
2399 {
2400   <*luatex>
2401   \tex_pdfvariable:D linkmargin
2402   </luatex>
2403   <*pdftex>
2404   \tex_pdflinkmargin:D
2405   </pdftex>
2406   \dim_eval:n {#1} \scan_stop:
2407 }

```

(End definition for `\_pdf_backend_link_margin:n`.)

`\_pdf_backend_destination:nn` A simple task: pass the data to the primitive. The `\scan_stop:` deals with the danger of an unterminated keyword. The zoom given here is a percentage, but we need to pass it as *per mille*. The rectangle version is also easy as everything is build in.

`\_pdf_backend_destination_box:nn`

```

2408 \cs_new_protected:Npn \_pdf_backend_destination:nn #1#2
2409 {
2410   <*luatex>
2411   \tex_pdfextension:D dest ~
2412   </luatex>
2413   <*pdftex>
2414   \tex_pdfdest:D
2415   </pdftex>
2416   name {#1}
2417   \str_case:nnF {#2}
2418   {
2419     { xyz } { xyz }
2420     { fit } { fit }
2421     { fitb } { fitb }
2422     { fitbh } { fitbh }
2423     { fitbv } { fitbv }
2424     { fith } { fith }
2425     { fitv } { fitv }
2426     { fitr } { fitr }
2427   }

```

```

2428         { xyz ~ zoom \fp_eval:n { #2 * 10 } }
2429     \scan_stop:
2430 }
2431 \cs_new_protected:Npn \__pdf_backend_destination_box:nn #1#2
2432 {
2433     \group_begin:
2434     \hbox_set:Nn \l__pdf_internal_box {#2}
2435     \*luatex
2436     \tex_pdfextension:D dest ~
2437     \*pdftex
2438     \tex_pdfdest:D
2439     \*pdftex
2440     name {#1}
2441     fitr ~
2442     width \box_wd:N \l__pdf_internal_box
2443     height \box_ht:N \l__pdf_internal_box
2444     depth \box_dp:N \l__pdf_internal_box
2445     \box_use:N \l__pdf_internal_box
2446     \group_end:
2447 }
2448
(End definition for \__pdf_backend_destination:nn and \__pdf_backend_destination_box:nn.)

```

### 6.3.2 Catalogue entries

```

\__pdf_backend_catalog_gput:nn
\__pdf_backend_info_gput:nn
2449 \cs_new_protected:Npn \__pdf_backend_catalog_gput:nn #1#2
2450 {
2451     \*luatex
2452     \tex_pdfextension:D catalog
2453     \*pdftex
2454     \tex_pdfcatalog:D
2455     { / #1 ~ #2 }
2456 }
2457 \cs_new_protected:Npn \__pdf_backend_info_gput:nn #1#2
2458 {
2459     \*luatex
2460     \tex_pdfextension:D info
2461     \*pdftex
2462     \tex_pdfinfo:D
2463     { / #1 ~ #2 }
2464 }
2465
(End definition for \__pdf_backend_catalog_gput:nn and \__pdf_backend_info_gput:nn.)

```

### 6.3.3 Objects

\g\_\_pdf\_backend\_object\_prop For tracking objects to allow finalisation.

```

2469 \prop_new:N \g__pdf_backend_object_prop

```



(End definition for `\g__pdf_backend_object_prop`.)

`\_pdf_backend_object_new:nn` Declaring objects means reserving at the PDF level plus starting tracking.

```

\__pdf_backend_object_ref:n
2470 \cs_new_protected:Npn \_pdf_backend_object_new:nn #1#2
2471 {
2472   \*luatex
2473   \tex_pdfextension:D obj ~
2474   \*pdfTeX
2475   \tex_pdfobj:D
2476   \*pdfTeX
2477   reserveobjnum ~
2478   \int_const:cn
2479   { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
2480   \*luatex
2481   { \tex_pdffeedback:D lastobj }
2482   \*pdfTeX
2483   { \tex_pdflastobj:D }
2484   \*pdfTeX
2485   \prop_gput:Nnn \g__pdf_backend_object_prop {#1} {#2}
2486   }
2487   \cs_new:Npn \_pdf_backend_object_ref:n #1
2488   { \int_use:c { c__pdf_backend_object_ \tl_to_str:n {#1} _int } ~ 0 ~ R }
2490

```

(End definition for `\_pdf_backend_object_new:nn` and `\_pdf_backend_object_ref:n`.)

`\_pdf_backend_object_write:nn` Writing the data needs a little information about the structure of the object.

```

\_pdf_backend_object_write:nx
\_pdf_exp_not_i:nn
\_pdf_exp_not_ii:nn
2491 \cs_new_protected:Npn \_pdf_backend_object_write:nn #1#2
2492 {
2493   \*luatex
2494   \tex_immediate:D \tex_pdfextension:D obj ~
2495   \*pdfTeX
2496   \tex_immediate:D \tex_pdfobj:D
2497   \*pdfTeX
2498   useobjnum ~
2499   \int_use:c
2500   { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
2501   \str_case_e:nn
2502   { \prop_item:Nn \g__pdf_backend_object_prop {#1} }
2503   {
2504     { array } { { [ ~ \exp_not:n {#2} ~ ] } }
2505     { dict } { { << ~ \exp_not:n {#2} ~ >> } }
2506     { fstream }
2507     {
2508       stream ~ attr ~ { \_pdf_exp_not_i:nn #2 } ~
2509       file ~ { \_pdf_exp_not_ii:nn #2 }
2510     }
2511   }
2512   { stream }
2513   {
2514     stream ~ attr ~ { \_pdf_exp_not_i:nn #2 } ~
2515     { \_pdf_exp_not_ii:nn #2 }
2516   }
2517 }

```

```

2517     }
2518   }
2519   \cs_generate_variant:Nn \__pdf_backend_object_write:nn { nx }
2520   \cs_new:Npn \__pdf_exp_not_i:nn #1#2 { \exp_not:n {#1} }
2521   \cs_new:Npn \__pdf_exp_not_ii:nn #1#2 { \exp_not:n {#2} }

(End definition for \__pdf_backend_object_write:nn, \__pdf_exp_not_i:nn, and \__pdf_exp_not_ii:nn.)

```

\\_\_pdf\_backend\_object\_now:nn Much like writing, but direct creation.

```

\__pdf_backend_object_now:nx
2522   \cs_new_protected:Npn \__pdf_backend_object_now:nn #1#2
2523   {
2524     \*luatex
2525     \tex_immediate:D \tex_pdfextension:D obj ~
2526     \*pdftex
2527     \tex_immediate:D \tex_pdfobj:D
2528     \*pdftex
2529     \str_case:nn
2530     {#1}
2531     {
2532       { array } { { [ ~ \exp_not:n {#2} ~ ] } }
2533       { dict } { { << ~ \exp_not:n {#2} ~ >> } }
2534       { fstream }
2535       {
2536         {
2537           stream ~ attr ~ { \__pdf_exp_not_i:nn #2 } ~
2538           file ~ { \__pdf_exp_not_ii:nn #2 }
2539         }
2540         { stream }
2541         {
2542           stream ~ attr ~ { \__pdf_exp_not_i:nn #2 } ~
2543           { \__pdf_exp_not_ii:nn #2 }
2544         }
2545       }
2546     }
2547   \cs_generate_variant:Nn \__pdf_backend_object_now:nn { nx }

(End definition for \__pdf_backend_object_now:nn.)

```

\\_\_pdf\_backend\_object\_last: Much like annotation.

```

2548   \cs_new:Npx \__pdf_backend_object_last:
2549   {
2550     \exp_not:N \int_value:w
2551     \*luatex
2552     \exp_not:N \tex_pdffeedback:D lastobj ~
2553     \*pdftex
2554     \exp_not:N \tex_pdflastobj:D
2555     \*pdftex
2556     \c_space_tl 0 ~ R
2557   }

(End definition for \__pdf_backend_object_last:.)

```

`\_pdf_backend_pageobject_ref:n` The usual wrapper situation; the three spaces here are essential.

```

2559 \cs_new:Npx \_pdf_backend_pageobject_ref:n #1
2560 {
2561   \exp_not:N \int_value:w
2562   <*luatex>
2563     \exp_not:N \tex_pdffeedback:D pageref
2564   </luatex>
2565   <*pdftex>
2566     \exp_not:N \tex_pdfpageref:D
2567   </pdftex>
2568     \c_space_tl #1 \c_space_tl \c_space_tl \c_space_tl 0 ~ R
2569 }

```

(End definition for `\_pdf_backend_pageobject_ref:n`.)

### 6.3.4 Structure

`\_pdf_backend_compresslevel:n` Simply pass data to the engine.

```

\_pdf_backend_compresslevel:n
\_pdf_backend_compress_objects:n
\_pdf_backend_objcompresslevel:n
2570 \cs_new_protected:Npn \_pdf_backend_compresslevel:n #1
2571 {
2572   \tex_global:D
2573   <*luatex>
2574     \tex_pdfvariable:D compresslevel
2575   </luatex>
2576   <*pdftex>
2577     \tex_pdfcompresslevel:D
2578   </pdftex>
2579     \int_value:w \int_eval:n {#1} \scan_stop:
2580 }
2581 \cs_new_protected:Npn \_pdf_backend_compress_objects:n #1
2582 {
2583   \bool_if:nTF {#1}
2584     { \_pdf_backend_objcompresslevel:n { 2 } }
2585     { \_pdf_backend_objcompresslevel:n { 0 } }
2586 }
2587 \cs_new_protected:Npn \_pdf_backend_objcompresslevel:n #1
2588 {
2589   \tex_global:D
2590   <*luatex>
2591     \tex_pdfvariable:D objcompresslevel
2592   </luatex>
2593   <*pdftex>
2594     \tex_pdfobjcompresslevel:D
2595   </pdftex>
2596     #1 \scan_stop:
2597 }

```

(End definition for `\_pdf_backend_compresslevel:n`, `\_pdf_backend_compress_objects:n`, and `\_pdf_backend_objcompresslevel:n`.)

`\_pdf_backend_version_major_gset:n` The availability of the primitive is not universal, so we have to test at load time.

```

\_pdf_backend_version_major_gset:n
\_pdf_backend_version_minor_gset:n
2598 \cs_new_protected:Npx \_pdf_backend_version_major_gset:n #1
2599 {
2600   <*luatex>

```

```

2601 \int_compare:nNnT \tex luatexversion:D > { 106 }
2602 {
2603   \exp_not:N \tex_global:D \tex_pdfvariable:D majorversion
2604   \exp_not:N \int_eval:n {#1} \scan_stop:
2605 }
2606 </luatex>
2607 <*pdftex>
2608 \cs_if_exist:NT \tex_pdfmajorversion:D
2609 {
2610   \exp_not:N \tex_global:D \tex_pdfmajorversion:D
2611   \exp_not:N \int_eval:n {#1} \scan_stop:
2612 }
2613 </pdftex>
2614 }
2615 \cs_new_protected:Npn \__pdf_backend_version_minor_gset:n #1
2616 {
2617   \tex_global:D
2618   <*luatex>
2619   \tex_pdfvariable:D minorversion
2620   </luatex>
2621   <*pdftex>
2622   \tex_pdfminorversion:D
2623   </pdftex>
2624   \int_eval:n {#1} \scan_stop:
2625 }

```

(End definition for \\_\_pdf\_backend\_version\_major\_gset:n and \\_\_pdf\_backend\_version\_minor\_gset:n.)

\\_\_pdf\_backend\_version\_major:  
 \\_\_pdf\_backend\_version\_minor:

As above.

```

2626 \cs_new:Npx \__pdf_backend_version_major:
2627 {
2628   <*luatex>
2629   \int_compare:nNnTF \tex luatexversion:D > { 106 }
2630   { \exp_not:N \tex_the:D \tex_pdfvariable:D majorversion }
2631   { 1 }
2632 </luatex>
2633 <*pdftex>
2634   \cs_if_exist:NTF \tex_pdfmajorversion:D
2635   { \exp_not:N \tex_the:D \tex_pdfmajorversion:D }
2636   { 1 }
2637 </pdftex>
2638 }
2639 \cs_new:Npn \__pdf_backend_version_minor:
2640 {
2641   \tex_the:D
2642   <*luatex>
2643   \tex_pdfvariable:D minorversion
2644   </luatex>
2645   <*pdftex>
2646   \tex_pdfminorversion:D
2647   </pdftex>
2648 }

```

(End definition for \\_\_pdf\_backend\_version\_major: and \\_\_pdf\_backend\_version\_minor:.)

### 6.3.5 Marked content

`\_pdf_backend_bdc:nn` Simple wrappers. May need refinement: see <https://chat.stackexchange.com/transcript/message/49970158#49970158>.  
`\_pdf_backend_emc:`

```
2649 \cs_new_protected:Npn \_pdf_backend_bdc:nn #1#2
2650 { \__kernel_backend_literal_page:n { /#1 ~ #2 ~ BDC } }
2651 \cs_new_protected:Npn \_pdf_backend_emc:
2652 { \__kernel_backend_literal_page:n { EMC } }
```

(End definition for `\_pdf_backend_bdc:nn` and `\_pdf_backend_emc:.`)

```
2653 </!latex | pdftex>
```

## 6.4 dvipdfmx backend

```
2654 <*dvipdfmx | xetex>
```

`\_pdf_backend:n` A generic function for the backend PDF specials: used where we can.

```
\_pdf_backend:x
2655 \cs_new_protected:Npx \_pdf_backend:n #1
2656 { \__kernel_backend_literal:n { pdf: #1 } }
2657 \cs_generate_variant:Nn \_pdf_backend:n { x }
```

(End definition for `\_pdf_backend:n`.)

### 6.4.1 Catalogue entries

```
\_pdf_backend_catalog_gput:nn
\_pdf_backend_info_gput:nn
2658 \cs_new_protected:Npn \_pdf_backend_catalog_gput:nn #1#2
2659 { \_pdf_backend:n { put ~ @catalog << /#1 ~ #2 >> } }
2660 \cs_new_protected:Npn \_pdf_backend_info_gput:nn #1#2
2661 { \_pdf_backend:n { docinfo << /#1 ~ #2 >> } }
```

(End definition for `\_pdf_backend_catalog_gput:nn` and `\_pdf_backend_info_gput:nn`.)

### 6.4.2 Objects

`\g__pdf_backend_object_int` For tracking objects to allow finalisation.  
`\g__pdf_backend_object_prop`

```
2662 \int_new:N \g__pdf_backend_object_int
2663 \prop_new:N \g__pdf_backend_object_prop
```

(End definition for `\g__pdf_backend_object_int` and `\g__pdf_backend_object_prop`.)

`\_pdf_backend_object_new:nn` Objects are tracked at the macro level, but we don't have to do anything at this stage.

```
\_pdf_backend_object_ref:n
2664 \cs_new_protected:Npn \_pdf_backend_object_new:nn #1#2
2665 {
2666   \int_gincr:N \g__pdf_backend_object_int
2667   \int_const:cn
2668   { c__pdf_backend_object_ \tl_to_str:n {#1} _int }
2669   { \g__pdf_backend_object_int }
2670   \prop_gput:Nnn \g__pdf_backend_object_prop {#1} {#2}
2671 }
2672 \cs_new:Npn \_pdf_backend_object_ref:n #1
2673 { @pdf.obj \int_use:c { c__pdf_backend_object_ \tl_to_str:n {#1} _int } }
```

(End definition for `\_pdf_backend_object_new:nn` and `\_pdf_backend_object_ref:n`.)

```

\__pdf_backend_object_write:nn
\__pdf_backend_object_write:nx
\__pdf_backend_object_write:nnn
\__pdf_backend_object_write_array:nn
\__pdf_backend_object_write_dict:nn
\__pdf_backend_object_write_fstream:nn
\__pdf_backend_object_write_stream:nn
\__pdf_backend_object_write_stream:nnnn

```

This is where we choose the actual type.

```

2674 \cs_new_protected:Npn \__pdf_backend_object_write:nn #1#2
2675 {
2676   \exp_args:Nx \__pdf_backend_object_write:nnn
2677   { \prop_item:Nn \g__pdf_backend_object_prop {#1} } {#1} {#2}
2678 }
2679 \cs_generate_variant:Nn \__pdf_backend_object_write:nn { nx }
2680 \cs_new_protected:Npn \__pdf_backend_object_write:nnn #1#2#3
2681 {
2682   \use:c { __pdf_backend_object_write_ #1 :nn }
2683   { \__pdf_backend_object_ref:n {#2} } {#3}
2684 }
2685 \cs_new_protected:Npn \__pdf_backend_object_write_array:nn #1#2
2686 {
2687   \__pdf_backend:x
2688   { obj ~ #1 ~ [ ~ \exp_not:n {#2} ~ ] }
2689 }
2690 \cs_new_protected:Npn \__pdf_backend_object_write_dict:nn #1#2
2691 {
2692   \__pdf_backend:x
2693   { obj ~ #1 ~ << ~ \exp_not:n {#2} ~ >> }
2694 }
2695 \cs_new_protected:Npn \__pdf_backend_object_write_fstream:nn #1#2
2696 { \__pdf_backend_object_write_stream:nnnn { f } {#1} #2 }
2697 \cs_new_protected:Npn \__pdf_backend_object_write_stream:nn #1#2
2698 { \__pdf_backend_object_write_stream:nnnn { } {#1} #2 }
2699 \cs_new_protected:Npn \__pdf_backend_object_write_stream:nnnn #1#2#3#4
2700 {
2701   \__pdf_backend:x
2702   {
2703     #1 stream ~ #2 ~
2704     ( \exp_not:n {#4} ) ~ << \exp_not:n {#3} >>
2705   }
2706 }

```

(End definition for \\_\_pdf\_backend\_object\_write:nn and others.)

```

\__pdf_backend_object_now:nn
\__pdf_backend_object_now:nx

```

No anonymous objects with dvipdfmx so we have to give an object name.

```

2707 \cs_new_protected:Npn \__pdf_backend_object_now:nn #1#2
2708 {
2709   \int_gincr:N \g__pdf_backend_object_int
2710   \exp_args:Nnx \use:c { __pdf_backend_object_write_ #1 :nn }
2711   { @pdf.obj \int_use:N \g__pdf_backend_object_int }
2712   {#2}
2713 }
2714 \cs_generate_variant:Nn \__pdf_backend_object_now:nn { nx }

```

(End definition for \\_\_pdf\_backend\_object\_now:nn.)

```

\__pdf_backend_object_last:

```

```

2715 \cs_new:Npn \__pdf_backend_object_last:
2716 { @pdf.obj \int_use:N \g__pdf_backend_object_int }

```

(End definition for \\_\_pdf\_backend\_object\_last:.)

`\_pdf_backend_pageobject_ref:n` Page references are easy in dvipdfmx/X<sub>Y</sub>TeX.

```

2717 \cs_new:Npn \_pdf_backend_pageobject_ref:n #1
2718 { @page #1 }

```

(End definition for `\_pdf_backend_pageobject_ref:n`.)

### 6.4.3 Annotations

`\g__pdf_landscape_bool` There is a bug in dvipdfmx/X<sub>Y</sub>TeX which means annotations do not rotate. As such, we need to know if landscape is active.

```

2719 \bool_new:N \g__pdf_landscape_bool
2720 \cs_if_exist:NT \landscape
2721 {
2722   \tl_put_right:Nn \landscape
2723     { \bool_gset_true:N \g__pdf_landscape_bool }
2724   \tl_put_left:Nn \endlandscape
2725     { \bool_gset_false:N \g__pdf_landscape_bool }
2726 }

```

(End definition for `\g__pdf_landscape_bool`.)

`\g_pdf_backend_annotation_int` Needed as objects which are not annotations could be created.

```

2727 \int_new:N \g_pdf_backend_annotation_int

```

(End definition for `\g_pdf_backend_annotation_int`.)

`\_pdf_backend_annotation:nnnn` Simply pass the raw data through, just dealing with evaluation of dimensions. The only wrinkle is landscape: we have to adjust by hand.

`\_pdf_backend_annotation_aux:nnnn`

```

2728 \cs_new_protected:Npn \_pdf_backend_annotation:nnnn #1#2#3#4
2729 {
2730   \bool_if:NTF \g__pdf_landscape_bool
2731   {
2732     \box_move_up:nn {#2}
2733     {
2734       \vbox:n
2735       {
2736         \_pdf_backend_annotation_aux:nnnn
2737           { #2 + #3 } {#1} { Opt } {#4}
2738       }
2739     }
2740   }
2741   { \_pdf_backend_annotation_aux:nnnn {#1} {#2} {#3} {#4} }
2742 }
2743 \cs_new_protected:Npn \_pdf_backend_annotation_aux:nnnn #1#2#3#4
2744 {
2745   \int_gincr:N \g_pdf_backend_object_int
2746   \int_gset_eq:NN \g_pdf_backend_annotation_int \g_pdf_backend_object_int
2747   \_pdf_backend:x
2748   {
2749     ann ~ @pdf.obj \int_use:N \g_pdf_backend_object_int \c_space_tl
2750     width ~ \dim_eval:n {#1} ~
2751     height ~ \dim_eval:n {#2} ~
2752     depth ~ \dim_eval:n {#3} ~
2753     <</Type/Annot #4 >>

```

```

2754     }
2755 }
(End definition for \_pdf_backend_annotation:nnnn and \_pdf_backend_annotation_aux:nnnn.)

```

\\_pdf\_backend\_annotation\_last:

```

2756 \cs_new:Npn \_pdf_backend_annotation_last:
2757 { @pdf.obj \int_use:N \g_pdf_backend_annotation_int }

```

(End definition for \\_pdf\_backend\_annotation\_last:.)

\g\_pdf\_backend\_link\_int

To track annotations which are links.

```

2758 \int_new:N \g_pdf_backend_link_int

```

(End definition for \g\_pdf\_backend\_link\_int.)

\\_pdf\_backend\_link\_begin\_goto:nnw

All created using the same internals.

\\_pdf\_backend\_link\_begin\_user:nnw

```

2759 \cs_new_protected:Npn \_pdf_backend_link_begin_goto:nnw #1#2

```

\\_pdf\_backend\_link\_begin:n

```

2760 { \_pdf_backend_link_begin:n { #1 /Subtype /Link /A << /S /GoTo /D ( #2 ) >> } }

```

\\_pdf\_backend\_link\_end:

```

2761 \cs_new_protected:Npn \_pdf_backend_link_begin_user:nnw #1#2

```

```

2762 { \_pdf_backend_link_begin:n {#1#2} }

```

```

2763 \cs_new_protected:Npx \_pdf_backend_link_begin:n #1

```

```

2764 {
2765   \int_compare:nNnF \c_kernel_sys_dvipdfmx_version_int < { 20201111 }
2766   {

```

```

2767     \exp_not:N \int_gincr:N \exp_not:N \g_pdf_backend_link_int
2768   }

```

```

2769   \_pdf_backend:x

```

```

2770   {
2771     bann ~
2772     \int_compare:nNnF \c_kernel_sys_dvipdfmx_version_int < { 20201111 }
2773     {

```

```

2774       @pdf.lnk
2775       \exp_not:N \int_use:N \exp_not:N \g_pdf_backend_link_int
2776       \c_space_tl
2777     }
2778     <<
2779     /Type /Annot
2780     #1
2781     >>
2782   }
2783 }

```

```

2784 \cs_new_protected:Npn \_pdf_backend_link_end:

```

```

2785 { \_pdf_backend:n { eann } }

```

(End definition for \\_pdf\_backend\_link\_begin\_goto:nnw and others.)

\\_pdf\_backend\_link\_last:

Available using the backend mechanism with a suitably-recent version.

```

2786 \cs_new:Npx \_pdf_backend_link_last:
2787 {
2788   \int_compare:nNnF \c_kernel_sys_dvipdfmx_version_int < { 20201111 }
2789   {
2790     @pdf.lnk
2791     \exp_not:N \int_use:N \exp_not:N \g_pdf_backend_link_int
2792   }
2793 }

```



(End definition for `\_pdf_backend_link_last:`.)

`\_pdf_backend_link_margin:n` Pass to dvipdfmx.

```
2794 \cs_new_protected:Npn \_pdf_backend_link_margin:n #1
2795 { \_kernel_backend_literal:x { dvipdfmx:config~g~ \dim_eval:n {#1} } }
```

(End definition for `\_pdf_backend_link_margin:n`.)

`\_pdf_backend_destination:nn`  
`\_pdf_backend_destination_box:nn`

Here, we need to turn the zoom into a scale. The method for `FitR` is from Alexander Grahn: the idea is to avoid needing to do any calculations in  $\text{\TeX}$  by using the backend data for `@xpos` and `@ypos`. `fitr` without rule spec doesn't work, so it falls back to `/Fit` here.

```
2796 \cs_new_protected:Npn \_pdf_backend_destination:nn #1#2
2797 {
2798   \_pdf_backend:x
2799   {
2800     dest ~ ( \exp_not:n {#1} )
2801     [
2802       @thispage
2803       \str_case:nnF {#2}
2804       {
2805         { xyz } { /XYZ ~ @xpos ~ @ypos ~ null }
2806         { fit } { /Fit }
2807         { fitb } { /FitB }
2808         { fitbh } { /FitBH }
2809         { fitbv } { /FitBV ~ @xpos }
2810         { fith } { /FitH ~ @ypos }
2811         { fitv } { /FitV ~ @xpos }
2812         { fitr } { /Fit }
2813       }
2814       { /XYZ ~ @xpos ~ @ypos ~ \fp_eval:n { (#2) / 100 } }
2815     ]
2816   }
2817 }
2818 \cs_new_protected:Npn \_pdf_backend_destination_box:nn #1#2
2819 {
2820   \group_begin:
2821   \hbox_set:Nn \l__pdf_internal_box {#2}
2822   \box_move_down:nn { \box_dp:N \l__pdf_internal_box }
2823   {
2824     \hbox:n
2825     {
2826       \_pdf_backend:n { obj ~ @pdf_ #1 _llx ~ @xpos }
2827       \_pdf_backend:n { obj ~ @pdf_ #1 _lly ~ @ypos }
2828     }
2829   }
2830   \box_use:N \l__pdf_internal_box
2831   \box_move_up:nn { \box_ht:N \l__pdf_internal_box }
2832   {
2833     \hbox:n
2834     {
2835       \_pdf_backend:n
2836       {
2837         dest ~ (#1)
```

```

2838         [
2839             @thispage
2840             /FitR ~
2841             @pdf_ #1 _llx ~ @pdf_ #1 _lly ~
2842             @xpos ~ @ypos
2843         ]
2844     }
2845 }
2846 }
2847 \group_end:
2848 }

```

(End definition for `\_pdf_backend_destination:nn` and `\_pdf_backend_destination_box:nn`.)

#### 6.4.4 Structure

`\_pdf_backend_compresslevel:n` Pass data to the backend: these are a one-shot.  
`\_pdf_backend_compress_objects:n`

```

2849 \cs_new_protected:Npn \_pdf_backend_compresslevel:n #1
2850 { \__kernel_backend_literal:x { dvipdfmx:config~z~ \int_eval:n {#1} } }
2851 \cs_new_protected:Npn \_pdf_backend_compress_objects:n #1
2852 {
2853     \bool_if:nF {#1}
2854     { \__kernel_backend_literal:n { dvipdfmx:config~C~0x40 } }
2855 }

```

(End definition for `\_pdf_backend_compresslevel:n` and `\_pdf_backend_compress_objects:n`.)

`\_pdf_backend_version_major_gset:n` We start with the assumption that the default is active.  
`\_pdf_backend_version_minor_gset:n`

```

2856 \cs_new_protected:Npn \_pdf_backend_version_major_gset:n #1
2857 {
2858     \cs_gset:Npx \_pdf_backend_version_major: { \int_eval:n {#1} }
2859     \__kernel_backend_literal:x { pdf:majorversion~ \_pdf_backend_version_major: }
2860 }
2861 \cs_new_protected:Npn \_pdf_backend_version_minor_gset:n #1
2862 {
2863     \cs_gset:Npx \_pdf_backend_version_minor: { \int_eval:n {#1} }
2864     \__kernel_backend_literal:x { pdf:minorversion~ \_pdf_backend_version_minor: }
2865 }

```

(End definition for `\_pdf_backend_version_major_gset:n` and `\_pdf_backend_version_minor_gset:n`.)

`\_pdf_backend_version_major:` We start with the assumption that the default is active.  
`\_pdf_backend_version_minor:`

```

2866 \cs_new:Npn \_pdf_backend_version_major: { 1 }
2867 \cs_new:Npn \_pdf_backend_version_minor: { 5 }

```

(End definition for `\_pdf_backend_version_major:` and `\_pdf_backend_version_minor:.`)

#### 6.4.5 Marked content

`\_pdf_backend_bdc:nn` Simple wrappers. May need refinement: see <https://chat.stackexchange.com/transcript/message/49970158#49970158>.  
`\_pdf_backend_emc:`

```

2868 \cs_new_protected:Npn \_pdf_backend_bdc:nn #1#2
2869 { \__kernel_backend_literal_page:n { /#1 ~ #2 ~ BDC } }
2870 \cs_new_protected:Npn \_pdf_backend_emc:
2871 { \__kernel_backend_literal_page:n { EMC } }

```

(End definition for `\_pdf_backend_bdc:nn` and `\_pdf_backend_emc:.`)

2872 `</dvipdfmx | xetex>`

## 6.5 dvisvgm backend

2873 `<*dvisvgm>`

### 6.5.1 Catalogue entries

No-op.

2874 `\cs_new_protected:Npn \_pdf_backend_catalog_gput:nn #1#2 { }`

2875 `\cs_new_protected:Npn \_pdf_backend_info_gput:nn #1#2 { }`

(End definition for `\_pdf_backend_catalog_gput:nn` and `\_pdf_backend_info_gput:nn`.)

### 6.5.2 Objects

All no-ops here.

2876 `\cs_new_protected:Npn \_pdf_backend_object_new:nn #1#2 { }`

2877 `\cs_new:Npn \_pdf_backend_object_ref:n #1 { }`

2878 `\cs_new_protected:Npn \_pdf_backend_object_write:nn #1#2 { }`

2879 `\cs_new_protected:Npn \_pdf_backend_object_write:nx #1#2 { }`

2880 `\cs_new_protected:Npn \_pdf_backend_object_now:nn #1#2 { }`

2881 `\cs_new_protected:Npn \_pdf_backend_object_now:nx #1#2 { }`

2882 `\cs_new:Npn \_pdf_backend_object_last: { }`

2883 `\cs_new:Npn \_pdf_backend_pageobject_ref:n #1 { }`

(End definition for `\_pdf_backend_object_new:nn` and others.)

### 6.5.3 Structure

These are all no-ops.

2884 `\cs_new_protected:Npn \_pdf_backend_compresslevel:n #1 { }`

2885 `\cs_new_protected:Npn \_pdf_backend_compress_objects:n #1 { }`

(End definition for `\_pdf_backend_compresslevel:n` and `\_pdf_backend_compress_objects:n`.)

Data not available!

2886 `\cs_new_protected:Npn \_pdf_backend_version_major_gset:n #1 { }`

2887 `\cs_new_protected:Npn \_pdf_backend_version_minor_gset:n #1 { }`

(End definition for `\_pdf_backend_version_major_gset:n` and `\_pdf_backend_version_minor_gset:n`.)

Data not available!

2888 `\cs_new:Npn \_pdf_backend_version_major: { -1 }`

2889 `\cs_new:Npn \_pdf_backend_version_minor: { -1 }`

(End definition for `\_pdf_backend_version_major:` and `\_pdf_backend_version_minor:.`)

More no-ops.

2890 `\cs_new_protected:Npn \_pdf_backend_bdc:nn #1#2 { }`

2891 `\cs_new_protected:Npn \_pdf_backend_emc: { }`

(End definition for `\_pdf_backend_bdc:nn` and `\_pdf_backend_emc:.`)

2892 `</dvisvgm>`

2893 `</package>`

## 7 l3backend-header Implementation

|                                 |                                                                                                                                                                                                                                                                |  |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                                 | 2894 <code>&lt;*dvips &amp; header&gt;</code>                                                                                                                                                                                                                  |  |
| <code>color.sc</code>           | Empty definitions for color at the top level.                                                                                                                                                                                                                  |  |
| <code>color.fc</code>           | 2895 <code>/color.sc { } def</code><br>2896 <code>/color.fc { } def</code>                                                                                                                                                                                     |  |
|                                 | <i>(End definition for color.sc and color.fc. These functions are documented on page ??.)</i>                                                                                                                                                                  |  |
| <code>TeXcolorseparation</code> | Support for separation/spot colors: this strange naming is so things work with the color                                                                                                                                                                       |  |
| <code>separation</code>         | stack.                                                                                                                                                                                                                                                         |  |
|                                 | 2897 <code>TeXDict begin</code><br>2898 <code>/TeXcolorseparation { setcolor } def</code><br>2899 <code>end</code>                                                                                                                                             |  |
|                                 | <i>(End definition for TeXcolorseparation and separation. These functions are documented on page ??.)</i>                                                                                                                                                      |  |
| <code>pdf.globaldict</code>     | A small global dictionary for backend use.                                                                                                                                                                                                                     |  |
|                                 | 2900 <code>true setglobal</code><br>2901 <code>/pdf.globaldict 4 dict def</code><br>2902 <code>false setglobal</code>                                                                                                                                          |  |
|                                 | <i>(End definition for pdf.globaldict. This function is documented on page ??.)</i>                                                                                                                                                                            |  |
| <code>pdf.cvs</code>            | Small utilities for PostScript manipulations. Conversion to DVI dimensions is done here                                                                                                                                                                        |  |
| <code>pdf.dvi.pt</code>         | to allow for Resolution. The total height of a rectangle (an array) needs a little maths,                                                                                                                                                                      |  |
| <code>pdf.pt.dvi</code>         | in contrast to simply extracting a value.                                                                                                                                                                                                                      |  |
| <code>pdf.rect.ht</code>        | 2903 <code>/pdf.cvs { 65534 string cvs } def</code><br>2904 <code>/pdf.dvi.pt { 72.27 mul Resolution div } def</code><br>2905 <code>/pdf.pt.dvi { 72.27 div Resolution mul } def</code><br>2906 <code>/pdf.rect.ht { dup 1 get neg exch 3 get add } def</code> |  |
|                                 | <i>(End definition for pdf.cvs and others. These functions are documented on page ??.)</i>                                                                                                                                                                     |  |
| <code>pdf.linkmargin</code>     | Settings which are defined up-front in SDict.                                                                                                                                                                                                                  |  |
| <code>pdf.linkdp.pad</code>     | 2907 <code>/pdf.linkmargin { 1 pdf.pt.dvi } def</code>                                                                                                                                                                                                         |  |
| <code>pdf.linkht.pad</code>     | 2908 <code>/pdf.linkdp.pad { 0 } def</code><br>2909 <code>/pdf.linkht.pad { 0 } def</code>                                                                                                                                                                     |  |
|                                 | <i>(End definition for pdf.linkmargin, pdf.linkdp.pad, and pdf.linkht.pad. These functions are documented on page ??.)</i>                                                                                                                                     |  |
| <code>pdf.rect</code>           | Functions for marking the limits of an annotation/link, plus drawing the border. We                                                                                                                                                                            |  |
| <code>pdf.save.ll</code>        | separate links for generic annotations to support adding a margin and setting a minimal                                                                                                                                                                        |  |
| <code>pdf.save.ur</code>        | size.                                                                                                                                                                                                                                                          |  |
| <code>pdf.save.linkll</code>    | 2910 <code>/pdf.rect</code>                                                                                                                                                                                                                                    |  |
| <code>pdf.save.linkur</code>    | 2911 <code>{ /Rect [ pdf.llx pdf.lly pdf.urx pdf.ury ] } def</code>                                                                                                                                                                                            |  |
| <code>pdf.llx</code>            | 2912 <code>/pdf.save.ll</code>                                                                                                                                                                                                                                 |  |
| <code>pdf.lly</code>            | 2913 <code>{</code>                                                                                                                                                                                                                                            |  |
| <code>pdf.urx</code>            | 2914 <code>currentpoint</code>                                                                                                                                                                                                                                 |  |
| <code>pdf.ury</code>            | 2915 <code>/pdf.lly exch def</code>                                                                                                                                                                                                                            |  |
|                                 | 2916 <code>/pdf.llx exch def</code>                                                                                                                                                                                                                            |  |
|                                 | 2917 <code>}</code>                                                                                                                                                                                                                                            |  |
|                                 | 2918 <code>def</code>                                                                                                                                                                                                                                          |  |

```

2919 /pdf.save.ur
2920 {
2921     currentpoint
2922     /pdf.ury exch def
2923     /pdf.urx exch def
2924 }
2925 def
2926 /pdf.save.linkll
2927 {
2928     currentpoint
2929     pdf.linkmargin add
2930     pdf.linkdp.pad add
2931     /pdf.lly exch def
2932     pdf.linkmargin sub
2933     /pdf.llx exch def
2934 }
2935 def
2936 /pdf.save.linkur
2937 {
2938     currentpoint
2939     pdf.linkmargin sub
2940     pdf.linkht.pad sub
2941     /pdf.ury exch def
2942     pdf.linkmargin add
2943     /pdf.urx exch def
2944 }
2945 def

```

*(End definition for pdf.rect and others. These functions are documented on page ??.)*

pdf.dest.anchor For finding the anchor point of a destination link. We make the use case a separate function as it comes up a lot, and as this makes it easier to adjust if we need additional effects. We also need a more complex approach to convert a co-ordinate pair correctly when defining a rectangle: this can otherwise be out when using a landscape page. pdf.dest.x pdf.dest.y pdf.dest.point pdf.dest2device (Thanks to Alexander Grahn for the approach here.)

```

pdf.dev.x 2946 /pdf.dest.anchor
pdf.dev.y 2947 {
pdf.tmpa 2948     currentpoint exch
pdf.tmpb 2949     pdf.dvi.pt 72 add
pdf.tmpc 2950     /pdf.dest.x exch def
pdf.tmpd 2951     pdf.dvi.pt
2952     vsize 72 sub exch sub
2953     /pdf.dest.y exch def
2954 }
2955 def
2956 /pdf.dest.point
2957 { pdf.dest.x pdf.dest.y } def
2958 /pdf.dest2device
2959 {
2960     /pdf.dest.y exch def
2961     /pdf.dest.x exch def
2962     matrix currentmatrix
2963     matrix defaultmatrix
2964     matrix invertmatrix

```

```

2965     matrix concatmatrix
2966     cvx exec
2967     /pdf.dev.y exch def
2968     /pdf.dev.x exch def
2969     /pdf.tmpd exch def
2970     /pdf.tmpc exch def
2971     /pdf.tmpb exch def
2972     /pdf.tmpa exch def
2973     pdf.dest.x pdf.tmpa mul
2974         pdf.dest.y pdf.tmpc mul add
2975         pdf.dev.x add
2976     pdf.dest.x pdf.tmpb mul
2977     pdf.dest.y pdf.tmpd mul add
2978     pdf.dev.y add
2979 }
2980 def

```

(End definition for pdf.dest.anchor and others. These functions are documented on page ??.)

|                             |                                                                                           |
|-----------------------------|-------------------------------------------------------------------------------------------|
| pdf.bordertracking          | To know where a breakable link can go, we need to track the boundary rectangle. That      |
| pdf.bordertracking.begin    | can be done by hooking into a and x operations: those names have to be retained. The      |
| pdf.bordertracking.end      | boundary is stored at the end of the operation. Special effort is needed at the start and |
| pdf.leftboundary            | end of pages (or rather galleys), such that everything works properly.                    |
| pdf.rightboundary           |                                                                                           |
| pdf.brokenlink.rect         | 2981 /pdf.bordertracking false def                                                        |
| pdf.brokenlink.skip         | 2982 /pdf.bordertracking.begin                                                            |
| pdf.brokenlink.dict         | 2983 {                                                                                    |
| pdf.bordertracking.endpage  | 2984 SDict /pdf.bordertracking true put                                                   |
| pdf.bordertracking.continue | 2985 SDict /pdf.leftboundary undef                                                        |
| pdf.originx                 | 2986 SDict /pdf.rightboundary undef                                                       |
| pdf.originy                 | 2987 /a where                                                                             |
|                             | 2988 {                                                                                    |
|                             | 2989 /a                                                                                   |
|                             | 2990 {                                                                                    |
|                             | 2991 currentpoint pop                                                                     |
|                             | 2992 SDict /pdf.rightboundary known dup                                                   |
|                             | 2993 {                                                                                    |
|                             | 2994 SDict /pdf.rightboundary get 2 index lt                                              |
|                             | 2995 { not }                                                                              |
|                             | 2996 if                                                                                   |
|                             | 2997 }                                                                                    |
|                             | 2998 if                                                                                   |
|                             | 2999 { pop }                                                                              |
|                             | 3000 { SDict exch /pdf.rightboundary exch put }                                           |
|                             | 3001 ifelse                                                                               |
|                             | 3002 moveto                                                                               |
|                             | 3003 currentpoint pop                                                                     |
|                             | 3004 SDict /pdf.leftboundary known dup                                                    |
|                             | 3005 {                                                                                    |
|                             | 3006 SDict /pdf.leftboundary get 2 index gt                                               |
|                             | 3007 { not }                                                                              |
|                             | 3008 if                                                                                   |
|                             | 3009 }                                                                                    |
|                             | 3010 if                                                                                   |
|                             | 3011 { pop }                                                                              |

```

3012         { SDict exch /pdf.leftboundary exch put }
3013     ifelse
3014 }
3015     put
3016 }
3017 if
3018 }
3019 def
3020 /pdf.bordertracking.end
3021 {
3022     /a where { /a { moveto } put } if
3023     /x where { /x { 0 exch rmoveto } put } if
3024     SDict /pdf.leftboundary known
3025     { pdf.outerbox 0 pdf.leftboundary put }
3026     if
3027     SDict /pdf.rightboundary known
3028     { pdf.outerbox 2 pdf.rightboundary put }
3029     if
3030     SDict /pdf.bordertracking false put
3031 }
3032 def
3033 /pdf.bordertracking.endpage
3034 {
3035     pdf.bordertracking
3036     {
3037         pdf.bordertracking.end
3038         true setglobal
3039         pdf.globaldict
3040         /pdf.brokenlink.rect [ pdf.outerbox aload pop ] put
3041         pdf.globaldict
3042         /pdf.brokenlink.skip pdf.baselineskip put
3043         pdf.globaldict
3044         /pdf.brokenlink.dict
3045         pdf.link.dict pdf.cvs put
3046         false setglobal
3047         mark pdf.link.dict cvx exec /Rect
3048         [
3049             pdf.llx
3050             pdf.lly
3051             pdf.outerbox 2 get pdf.linkmargin add
3052             currentpoint exch pop
3053             pdf.outerbox pdf.rect.ht sub pdf.linkmargin sub
3054         ]
3055         /ANN pdf.pdfmark
3056     }
3057     if
3058 }
3059 def
3060 /pdf.bordertracking.continue
3061 {
3062     /pdf.link.dict pdf.globaldict
3063     /pdf.brokenlink.dict get def
3064     /pdf.outerbox pdf.globaldict
3065     /pdf.brokenlink.rect get def

```

```

3066 /pdf.baselineskip pdf.globaldict
3067 /pdf.brokenlink.skip get def
3068 pdf.globaldict dup dup
3069 /pdf.brokenlink.dict undef
3070 /pdf.brokenlink.skip undef
3071 /pdf.brokenlink.rect undef
3072 currentpoint
3073 /pdf.originy exch def
3074 /pdf.originx exch def
3075 /a where
3076 {
3077   /a
3078   {
3079     moveto
3080     SDict
3081     begin
3082       currentpoint pdf.originy ne exch
3083       pdf.originx ne or
3084       {
3085         pdf.save.linkll
3086         /pdf.lly
3087         pdf.lly pdf.outerbox 1 get sub def
3088         pdf.bordertracking.begin
3089       }
3090       if
3091       end
3092     }
3093     put
3094   }
3095   if
3096   /x where
3097   {
3098     /x
3099     {
3100       0 exch rmoveto
3101       SDict
3102       begin
3103         currentpoint
3104         pdf.originy ne exch pdf.originx ne or
3105         {
3106           pdf.save.linkll
3107           /pdf.lly
3108           pdf.lly pdf.outerbox 1 get sub def
3109           pdf.bordertracking.begin
3110         }
3111         if
3112         end
3113       }
3114       put
3115     }
3116     if
3117   }
3118   def

```

(End definition for pdf.bordertracking and others. These functions are documented on page ??.)



|                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>pdf.breaklink</p> <p>pdf.breaklink.write</p> <p>pdf.count</p> <p>pdf.currentrect</p> | <p>Dealing with link breaking itself has multiple stage. The first step is to find the <b>Rect</b> entry in the dictionary, looping over key-value pairs. The first line is handled first, adjusting the rectangle to stay inside the text area. The second phase is a loop over the height of the bulk of the link area, done on the basis of a number of baselines. Finally, the end of the link area is tidied up, again from the boundary of the text area.</p> |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

3119 /pdf.breaklink
3120 {
3121     pop
3122     counttomark 2 mod 0 eq
3123     {
3124         counttomark /pdf.count exch def
3125         {
3126             pdf.count 0 eq { exit } if
3127             counttomark 2 roll
3128             1 index /Rect eq
3129             {
3130                 dup 4 array copy
3131                 dup dup
3132                 1 get
3133                 pdf.outerbox pdf.rect.ht
3134                 pdf.linkmargin 2 mul add sub
3135                 3 exch put
3136                 dup
3137                 pdf.outerbox 2 get
3138                 pdf.linkmargin add
3139                 2 exch put
3140                 dup dup
3141                 3 get
3142                 pdf.outerbox pdf.rect.ht
3143                 pdf.linkmargin 2 mul add add
3144                 1 exch put
3145                 /pdf.currentrect exch def
3146                 pdf.breaklink.write
3147                 {
3148                     pdf.currentrect
3149                     dup
3150                     pdf.outerbox 0 get
3151                     pdf.linkmargin sub
3152                     0 exch put
3153                     dup
3154                     pdf.outerbox 2 get
3155                     pdf.linkmargin add
3156                     2 exch put
3157                     dup dup
3158                     1 get
3159                     pdf.baselineskip add
3160                     1 exch put
3161                     dup dup
3162                     3 get
3163                     pdf.baselineskip add
3164                     3 exch put
3165                     /pdf.currentrect exch def
3166                     pdf.breaklink.write

```

```

3167         }
3168         1 index 3 get
3169         pdf.linkmargin 2 mul add
3170         pdf.outerbox pdf.rect.ht add
3171         2 index 1 get sub
3172         pdf.baselineskip div round cvi 1 sub
3173         exch
3174         repeat
3175         pdf.currentrect
3176         dup
3177         pdf.outerbox 0 get
3178         pdf.linkmargin sub
3179         0 exch put
3180         dup dup
3181         1 get
3182         pdf.baselineskip add
3183         1 exch put
3184         dup dup
3185         3 get
3186         pdf.baselineskip add
3187         3 exch put
3188         dup 2 index 2 get 2 exch put
3189         /pdf.currentrect exch def
3190         pdf.breaklink.write
3191         SDict /pdf.pdfmark.good false put
3192         exit
3193     }
3194     { pdf.count 2 sub /pdf.count exch def }
3195     ifelse
3196 }
3197 loop
3198 }
3199 if
3200 /ANN
3201 }
3202 def
3203 /pdf.breaklink.write
3204 {
3205     counttomark 1 sub
3206     index /_objdef eq
3207     {
3208         counttomark -2 roll
3209         dup wcheck
3210         {
3211             readonly
3212             counttomark 2 roll
3213         }
3214         { pop pop }
3215     } ifelse
3216 }
3217 if
3218 counttomark 1 add copy
3219 pop pdf.currentrect
3220 /ANN pdfmark

```

```

3221 }
3222 def

```

*(End definition for pdf.breaklink and others. These functions are documented on page ??.)*

|                  |                                                                                            |
|------------------|--------------------------------------------------------------------------------------------|
| pdf.pdfmark      | The business end of breaking links starts by hooking into pdfmarks. Unlike hypdvips,       |
| pdf.pdfmark.good | we avoid altering any links we have not created by using a copy of the core pdfmarks       |
| pdf.outerbox     | function. Only mark types which are known are altered. At present, this is purely ANN      |
| pdf.baselineskip | marks, which are measured relative to the size of the baseline skip. If they are more than |
| pdf.pdfmark.dict | one apparent line high, breaking is applied.                                               |

```

3223 /pdf.pdfmark
3224 {
3225     SDict /pdf.pdfmark.good true put
3226     dup /ANN eq
3227     {
3228         pdf.pdfmark.store
3229         pdf.pdfmark.dict
3230         begin
3231             Subtype /Link eq
3232             currentdict /Rect known and
3233             SDict /pdf.outerbox known and
3234             SDict /pdf.baselineskip known and
3235             {
3236                 Rect 3 get
3237                 pdf.linkmargin 2 mul add
3238                 pdf.outerbox pdf.rect.ht add
3239                 Rect 1 get sub
3240                 pdf.baselineskip div round cvi 0 gt
3241                 { pdf.breaklink }
3242                 if
3243             }
3244             if
3245             end
3246             SDict /pdf.outerbox undef
3247             SDict /pdf.baselineskip undef
3248             currentdict /pdf.pdfmark.dict undef
3249         }
3250         if
3251         pdf.pdfmark.good
3252         { pdfmark }
3253         { cleartomark }
3254         ifelse
3255     }
3256     def
3257 /pdf.pdfmark.store
3258 {
3259     /pdf.pdfmark.dict 65534 dict def
3260     counttomark 1 add copy
3261     pop
3262     {
3263         dup mark eq
3264         {
3265             pop
3266             exit

```

```

3267     }
3268     {
3269         pdf.pdfmark.dict
3270         begin def end
3271     }
3272     ifelse
3273 }
3274 loop
3275 }
3276 def

```

*(End definition for pdf.pdfmark and others. These functions are documented on page ??.)*

```

3277 </dvips & header>

```

# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

- \\_ . . . . . 146
- A**
- \AtBeginDvi . . . . . 59, 60, 636, 637
- B**
- bool commands:
  - \bool\_gset\_false:N . . . . . 1083, 1102, 1128, 1151, 1167, 1268, 1505, 1541, 2106, 2152, 2725
  - \bool\_gset\_true:N . . . . . 1081, 1154, 1266, 1520, 2099, 2105, 2723
  - \bool\_if:NTF . . . . . 57, 634, 1093, 1097, 1115, 1119, 1123, 1137, 1142, 1146, 1158, 1162, 1279, 1284, 1289, 1479, 1524, 1637, 1672, 1782, 1824, 2094, 2109, 2114, 2119, 2730
  - \bool\_if:nTF . . . . . 2319, 2583, 2853
  - \bool\_lazy\_or:nnTF . . . . . 1664, 1817
  - \bool\_new:N . . . . . 1084, 1155, 1269, 1521, 2082, 2083, 2719
  - \bool\_set\_false:N . . . . . 1647, 1749, 1842, 1906
- box commands:
  - \box\_dp:N . . . . . 216, 218, 266, 268, 323, 325, 372, 374, 376, 378, 2131, 2164, 2165, 2190, 2286, 2445, 2822
  - \box\_ht:N . . . . . 218, 268, 325, 376, 378, 1684, 1879, 2136, 2175, 2176, 2192, 2290, 2444, 2831
  - \box\_if\_empty:N . . . . . 2225
  - \box\_move\_down:nn . . . . . 2056, 2131, 2285, 2822
  - \box\_move\_up:nn . . . . . 2058, 2136, 2289, 2732, 2831
  - \box\_new:N . . . . . 1941, 2046, 2047
  - \box\_set\_dp:Nn . . . . . 1604
  - \box\_set\_ht:Nn . . . . . 1603
  - \box\_set\_wd:Nn . . . . . 280, 1602
  - \box\_use:N . . . . . 223, 241, 255, 271, 298, 312, 328, 344, 356, 407, 421, 440, 1219, 1414, 1605, 2087, 2288, 2446, 2830
  - \box\_wd:N . . . . . 217, 225, 267, 273, 324, 330, 373, 375, 1683, 1878, 2443
- box internal commands:
  - \\_\_box\_backend\_clip:N . . . . . 205, 260, 317, 361
  - \\_\_box\_backend\_cos\_fp . . . . . 275
  - \\_\_box\_backend\_rotate:Nn . . . . . 227, 275, 332, 411
  - \\_\_box\_backend\_rotate\_aux:Nn . . . . . 227, 275, 332
  - \\_\_box\_backend\_scale:Nnn . . . . . 244, 303, 347, 424
  - \l\_\_box\_backend\_sin\_fp . . . . . 275
  - \g\_\_box\_clip\_path\_int . . . . . 361
- C**
- char commands:
  - \char\_set\_catcode\_space:n . . . . . 146
- clist commands:
  - \clist\_map\_function:nN . . . . . 1175, 1299
  - \clist\_map\_function:nn . . . . . 1548
- color internal commands:
  - \\_\_color\_backend:nnn . . . . . 988
  - \\_\_color\_backend:nnnn . . . . . 958
  - \\_\_color\_backend\_cmyk:nw . . . . . 958
  - \\_\_color\_backend\_devicen\_init:n . . . . . 868
  - \\_\_color\_backend\_devicen\_init:nnn . . . . . 782, 868
  - \\_\_color\_backend\_devicen\_init:w . . . . . 868
  - \\_\_color\_backend\_fill\_cmyk:n . . . . . 918, 938, 958
  - \\_\_color\_backend\_fill\_devicen:nn . . . . . 930, 950, 1020
  - \\_\_color\_backend\_fill\_gray:n . . . . . 918, 938, 958
  - \\_\_color\_backend\_fill\_rgb:n . . . . . 918, 938, 958
  - \\_\_color\_backend\_fill\_separation:nn . . . . . 930, 950, 1020
  - \\_\_color\_backend\_grab:nn . . . . . 975, 977
  - \\_\_color\_backend\_gray:nn . . . . . 958
  - \\_\_color\_backend\_gray\_aux:n . . . . . 958
  - \\_\_color\_backend\_gray\_aux:nn . . . . . 982, 987
  - \\_\_color\_backend\_pickup:N . . . . . 448, 471
  - \\_\_color\_backend\_pickup:w . . . . . 14, 448, 471
  - \\_\_color\_backend\_reset: . . . . . 580, 600, 615
  - \\_\_color\_backend\_rgb:nw . . . . . 958
  - \\_\_color\_backend\_select:n . . . . . 580, 601, 603, 605, 606, 630, 808
  - \\_\_color\_backend\_select\_cmyk:n . . . . . 580, 600, 615
  - \\_\_color\_backend\_select\_devicen:nn . . . . . 629, 802, 808

|                                                                                                                       |                                                                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\__color_backend_select_gray:n</code> ..                                                                        | cs commands:                                                                                                                                                                |
| ..... <a href="#">580</a> , <a href="#">600</a> , <a href="#">615</a>                                                 | <code>\cs_generate_variant:Nn</code> .....                                                                                                                                  |
| <code>\__color_backend_select_rgb:n</code> ...                                                                        | ..... <a href="#">49</a> , <a href="#">53</a> , <a href="#">56</a> , <a href="#">91</a> , <a href="#">130</a> ,                                                             |
| ..... <a href="#">580</a> , <a href="#">600</a> , <a href="#">615</a>                                                 | <a href="#">135</a> , <a href="#">162</a> , <a href="#">193</a> , <a href="#">199</a> , <a href="#">645</a> , <a href="#">1030</a> , <a href="#">1229</a> ,                 |
| <code>\__color_backend_select_separation:nn</code>                                                                    | <a href="#">1423</a> , <a href="#">1796</a> , <a href="#">1853</a> , <a href="#">1869</a> , <a href="#">1945</a> , <a href="#">1982</a> ,                                   |
| ..... <a href="#">629</a> , <a href="#">802</a> , <a href="#">808</a>                                                 | <a href="#">2041</a> , <a href="#">2519</a> , <a href="#">2547</a> , <a href="#">2657</a> , <a href="#">2679</a> , <a href="#">2714</a>                                     |
| <code>\__color_backend_separation-</code>                                                                             | <code>\cs_gset:Npx</code> .....                                                                                                                                             |
| <code>init:n</code> .....                                                                                             | <a href="#">2858</a> , <a href="#">2863</a>                                                                                                                                 |
| ..... <a href="#">632</a> , <a href="#">811</a> , <a href="#">892</a> , <a href="#">915</a>                           | <code>\cs_gset_eq:NN</code> .....                                                                                                                                           |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">622</a> , <a href="#">623</a>                                                                                                                                   |
| <code>init:nnn</code> .....                                                                                           | <code>\cs_gset_protected:Npn</code> .....                                                                                                                                   |
| ..... <a href="#">632</a>                                                                                             | ..... <a href="#">617</a> , <a href="#">624</a> , <a href="#">836</a> , <a href="#">865</a> , <a href="#">908</a>                                                           |
| <code>\__color_backend_separation-</code>                                                                             | <code>\cs_if_exist:NTF</code> .....                                                                                                                                         |
| <code>init:nnnn</code> .....                                                                                          | ..... <a href="#">27</a> , <a href="#">59</a> , <a href="#">449</a> , <a href="#">472</a> , <a href="#">636</a> ,                                                           |
| ..... <a href="#">632</a>                                                                                             | <a href="#">835</a> , <a href="#">863</a> , <a href="#">907</a> , <a href="#">2221</a> , <a href="#">2608</a> , <a href="#">2634</a> , <a href="#">2720</a>                 |
| <code>\__color_backend_separation-</code>                                                                             | <code>\cs_if_exist_use:NTF</code> .....                                                                                                                                     |
| <code>init:nnnnn</code> .....                                                                                         | <a href="#">38</a> , <a href="#">658</a>                                                                                                                                    |
| ..... <a href="#">632</a> , <a href="#">804</a> , <a href="#">811</a>                                                 | <code>\cs_new:Npn</code> .....                                                                                                                                              |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">667</a> , <a href="#">669</a> , <a href="#">671</a> ,                                                                                                           |
| <code>init:nw</code> .....                                                                                            | <a href="#">673</a> , <a href="#">680</a> , <a href="#">686</a> , <a href="#">688</a> , <a href="#">694</a> , <a href="#">711</a> , <a href="#">718</a> ,                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">720</a> , <a href="#">909</a> , <a href="#">1180</a> , <a href="#">1304</a> , <a href="#">1552</a> , <a href="#">1882</a> ,                                     |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">1891</a> , <a href="#">1935</a> , <a href="#">1960</a> , <a href="#">2042</a> , <a href="#">2044</a> , <a href="#">2077</a> ,                                   |
| <code>init:w</code> .....                                                                                             | <a href="#">2246</a> , <a href="#">2331</a> , <a href="#">2332</a> , <a href="#">2489</a> , <a href="#">2520</a> , <a href="#">2521</a> ,                                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">2639</a> , <a href="#">2672</a> , <a href="#">2715</a> , <a href="#">2717</a> , <a href="#">2756</a> , <a href="#">2866</a> ,                                   |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">2867</a> , <a href="#">2877</a> , <a href="#">2882</a> , <a href="#">2883</a> , <a href="#">2888</a> , <a href="#">2889</a>                                     |
| <code>init_/DeviceCMYK:nnn</code> .....                                                                               | <code>\cs_new:Npx</code> .....                                                                                                                                              |
| ..... <a href="#">632</a>                                                                                             | .. <a href="#">2352</a> , <a href="#">2387</a> , <a href="#">2548</a> , <a href="#">2559</a> , <a href="#">2626</a> , <a href="#">2786</a>                                  |
| <code>\__color_backend_separation-</code>                                                                             | <code>\cs_new_eq:NN</code> .....                                                                                                                                            |
| <code>init_/DeviceGray:nnn</code> .....                                                                               | <a href="#">46</a> , <a href="#">631</a> , <a href="#">810</a> ,                                                                                                            |
| ..... <a href="#">632</a>                                                                                             | <a href="#">915</a> , <a href="#">934</a> , <a href="#">935</a> , <a href="#">954</a> , <a href="#">955</a> , <a href="#">1022</a> , <a href="#">1023</a> ,                 |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">1029</a> , <a href="#">1228</a> , <a href="#">1234</a> , <a href="#">1235</a> , <a href="#">1422</a> , <a href="#">1429</a> ,                                   |
| <code>init_/DeviceRGB:nnn</code> .....                                                                                | <a href="#">1614</a> , <a href="#">1643</a> , <a href="#">1694</a> , <a href="#">1695</a> , <a href="#">1737</a> , <a href="#">1745</a> ,                                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">1767</a> , <a href="#">1838</a> , <a href="#">1895</a> , <a href="#">1902</a> , <a href="#">1934</a> , <a href="#">2087</a>                                     |
| <code>\__color_backend_separation-</code>                                                                             | <code>\cs_new_protected:Npn</code> .....                                                                                                                                    |
| <code>init_aux:nnnnn</code> .....                                                                                     | ..... <a href="#">47</a> , <a href="#">51</a> , <a href="#">54</a> , <a href="#">64</a> , <a href="#">70</a> ,                                                              |
| ..... <a href="#">632</a>                                                                                             | <a href="#">75</a> , <a href="#">77</a> , <a href="#">81</a> , <a href="#">92</a> , <a href="#">102</a> , <a href="#">111</a> , <a href="#">120</a> , <a href="#">133</a> , |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">136</a> , <a href="#">138</a> , <a href="#">140</a> , <a href="#">160</a> , <a href="#">165</a> , <a href="#">174</a> , <a href="#">184</a> ,                   |
| <code>init_CIELAB:nnn</code> ....                                                                                     | <a href="#">194</a> , <a href="#">205</a> , <a href="#">227</a> , <a href="#">229</a> , <a href="#">244</a> , <a href="#">260</a> , <a href="#">275</a> ,                   |
| ..... <a href="#">632</a> , <a href="#">804</a> , <a href="#">811</a>                                                 | <a href="#">277</a> , <a href="#">303</a> , <a href="#">317</a> , <a href="#">332</a> , <a href="#">334</a> , <a href="#">347</a> , <a href="#">361</a> ,                   |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">411</a> , <a href="#">424</a> , <a href="#">448</a> , <a href="#">466</a> , <a href="#">471</a> , <a href="#">479</a> , <a href="#">508</a> ,                   |
| <code>init_CIELAB:nnnnnn</code> .....                                                                                 | <a href="#">523</a> , <a href="#">532</a> , <a href="#">544</a> , <a href="#">558</a> , <a href="#">568</a> , <a href="#">580</a> , <a href="#">582</a> ,                   |
| ..... <a href="#">805</a>                                                                                             | <a href="#">584</a> , <a href="#">586</a> , <a href="#">595</a> , <a href="#">600</a> , <a href="#">602</a> , <a href="#">604</a> , <a href="#">606</a> ,                   |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">611</a> , <a href="#">629</a> , <a href="#">646</a> , <a href="#">736</a> , <a href="#">782</a> , <a href="#">802</a> , <a href="#">803</a> ,                   |
| <code>init_count:n</code> .....                                                                                       | <a href="#">804</a> , <a href="#">805</a> , <a href="#">808</a> , <a href="#">811</a> , <a href="#">837</a> , <a href="#">841</a> , <a href="#">868</a> ,                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">918</a> , <a href="#">920</a> , <a href="#">922</a> , <a href="#">924</a> , <a href="#">926</a> , <a href="#">928</a> , <a href="#">930</a> ,                   |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">932</a> , <a href="#">938</a> , <a href="#">940</a> , <a href="#">942</a> , <a href="#">944</a> , <a href="#">946</a> , <a href="#">948</a> ,                   |
| <code>init_count:w</code> .....                                                                                       | <a href="#">950</a> , <a href="#">952</a> , <a href="#">958</a> , <a href="#">960</a> , <a href="#">962</a> , <a href="#">974</a> , <a href="#">976</a> ,                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">978</a> , <a href="#">987</a> , <a href="#">989</a> , <a href="#">991</a> , <a href="#">993</a> , <a href="#">1020</a> , <a href="#">1021</a> ,                 |
| <code>\__color_backend_separation-</code>                                                                             | <a href="#">1031</a> , <a href="#">1036</a> , <a href="#">1041</a> , <a href="#">1043</a> , <a href="#">1045</a> , <a href="#">1053</a> ,                                   |
| <code>init_Device:Nn</code> .....                                                                                     | <a href="#">1061</a> , <a href="#">1070</a> , <a href="#">1080</a> , <a href="#">1082</a> , <a href="#">1085</a> , <a href="#">1087</a> ,                                   |
| ..... <a href="#">632</a>                                                                                             | <a href="#">1104</a> , <a href="#">1109</a> , <a href="#">1130</a> , <a href="#">1153</a> , <a href="#">1156</a> , <a href="#">1169</a> ,                                   |
| <code>\__color_backend_stroke_cmyk:n</code> ..                                                                        | <a href="#">1182</a> , <a href="#">1187</a> , <a href="#">1189</a> , <a href="#">1191</a> , <a href="#">1193</a> , <a href="#">1195</a> ,                                   |
| ..... <a href="#">918</a> , <a href="#">938</a> , <a href="#">958</a>                                                 | <a href="#">1197</a> , <a href="#">1199</a> , <a href="#">1201</a> , <a href="#">1206</a> , <a href="#">1230</a> , <a href="#">1232</a> ,                                   |
| <code>\__color_backend_stroke_devicen:nn</code>                                                                       | <a href="#">1236</a> , <a href="#">1241</a> , <a href="#">1246</a> , <a href="#">1256</a> , <a href="#">1265</a> , <a href="#">1267</a> ,                                   |
| ..... <a href="#">930</a> , <a href="#">950</a> , <a href="#">1020</a>                                                | <a href="#">1270</a> , <a href="#">1272</a> , <a href="#">1274</a> , <a href="#">1276</a> , <a href="#">1281</a> , <a href="#">1286</a> ,                                   |
| <code>\__color_backend_stroke_gray:n</code> ..                                                                        | <a href="#">1291</a> , <a href="#">1293</a> , <a href="#">1306</a> , <a href="#">1311</a> , <a href="#">1313</a> , <a href="#">1315</a> ,                                   |
| ..... <a href="#">918</a> , <a href="#">938</a> , <a href="#">958</a>                                                 | <a href="#">1317</a> , <a href="#">1319</a> , <a href="#">1321</a> , <a href="#">1323</a> , <a href="#">1325</a> , <a href="#">1336</a> ,                                   |
| <code>\__color_backend_stroke_rgb:n</code> ...                                                                        | <a href="#">1361</a> , <a href="#">1373</a> , <a href="#">1385</a> , <a href="#">1397</a> , <a href="#">1404</a> , <a href="#">1424</a> ,                                   |
| ..... <a href="#">918</a> , <a href="#">938</a> , <a href="#">958</a>                                                 | <a href="#">1430</a> , <a href="#">1435</a> , <a href="#">1440</a> , <a href="#">1451</a> , <a href="#">1461</a> , <a href="#">1471</a> ,                                   |
| <code>\__color_backend_stroke_separation:nn</code>                                                                    |                                                                                                                                                                             |
| ..... <a href="#">930</a> , <a href="#">950</a> , <a href="#">1020</a>                                                |                                                                                                                                                                             |
| <code>\g_color_model_int</code> <a href="#">652</a> , <a href="#">788</a> , <a href="#">831</a> , <a href="#">903</a> |                                                                                                                                                                             |
| <code>\c_color_model_range_CIELAB_tl</code> .                                                                         |                                                                                                                                                                             |
| ..... <a href="#">743</a> , <a href="#">778</a> , <a href="#">852</a> , <a href="#">859</a>                           |                                                                                                                                                                             |
| <code>\g_color_stack_int</code> .....                                                                                 |                                                                                                                                                                             |
| <a href="#">505</a>                                                                                                   |                                                                                                                                                                             |
| <code>color.fc</code> .....                                                                                           |                                                                                                                                                                             |
| <a href="#">580</a> , <a href="#">2895</a>                                                                            |                                                                                                                                                                             |
| <code>color.sc</code> .....                                                                                           |                                                                                                                                                                             |
| <a href="#">580</a> , <a href="#">2895</a>                                                                            |                                                                                                                                                                             |

1473, 1475, 1477, 1508, 1510, 1515,  
 1517, 1519, 1522, 1543, 1554, 1567,  
 1569, 1571, 1573, 1575, 1577, 1579,  
 1581, 1583, 1591, 1615, 1629, 1644,  
 1656, 1661, 1689, 1701, 1714, 1724,  
 1739, 1746, 1754, 1765, 1769, 1772,  
 1787, 1797, 1832, 1839, 1845, 1851,  
 1854, 1861, 1870, 1875, 1883, 1896,  
 1903, 1909, 1911, 1913, 1924, 1943,  
 1946, 1948, 1952, 1962, 1983, 1988,  
 1993, 1998, 2008, 2013, 2021, 2049,  
 2054, 2086, 2088, 2090, 2092, 2097,  
 2112, 2117, 2154, 2183, 2202, 2211,  
 2248, 2255, 2281, 2305, 2317, 2329,  
 2330, 2333, 2335, 2339, 2363, 2365,  
 2367, 2378, 2398, 2408, 2431, 2449,  
 2459, 2470, 2491, 2522, 2570, 2581,  
 2587, 2615, 2649, 2651, 2658, 2660,  
 2664, 2674, 2680, 2685, 2690, 2695,  
 2697, 2699, 2707, 2728, 2743, 2759,  
 2761, 2784, 2794, 2796, 2818, 2849,  
 2851, 2856, 2861, 2868, 2870, 2874,  
 2875, 2876, 2878, 2879, 2880, 2881,  
 2884, 2885, 2886, 2887, 2890, 2891  
 \cs\_new\_protected:Npx .....  
 ..... 632, 1005, 2598, 2655, 2763  
 \cs\_set:Npn ..... 144  
 \cs\_set\_eq:NN ..... 2242, 2243  
 \cs\_set\_protected:Npn ..... 451, 474

## D

dim commands:

\dim\_eval:n ..... 2052, 2347, 2348,  
 2349, 2406, 2750, 2751, 2752, 2795  
 \dim\_max:nn ..... 2162, 2173  
 \dim\_set:Nn ... 1683, 1684, 1878, 1879  
 \dim\_to\_decimal:n .. 372, 373, 374,  
 375, 376, 378, 1433, 1438, 1444,  
 1445, 1446, 1447, 1456, 1457, 1458,  
 1549, 1568, 1929, 1930, 2160, 2171,  
 2189, 2190, 2191, 2192, 2196, 2252  
 \dim\_to\_decimal\_in\_bp:n .....  
 .... 216, 217, 218, 266, 267, 268,  
 323, 324, 325, 1049, 1050, 1057,  
 1058, 1065, 1066, 1074, 1075, 1076,  
 1177, 1181, 1185, 1239, 1244, 1250,  
 1251, 1252, 1260, 1261, 1301, 1305,  
 1309, 1553, 1620, 1621, 1622, 1623,  
 1759, 1760, 1761, 1762, 1811, 1812,  
 1813, 1814, 1918, 1919, 1920, 1921

draw internal commands:

\\_\_draw\_align\_currentpoint:... .. 32  
 \\_\_draw\_backend\_add\_to\_path:n ...  
 ..... 1430, 1476

\\_\_draw\_backend\_begin: .....  
 ..... 1031, 1230, 1424  
 \\_\_draw\_backend\_box\_use:Nnnnn ...  
 ..... 28, 1206, 1404, 1591  
 \\_\_draw\_backend\_cap\_but: .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_cap\_rectangle: ..  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_cap\_round: .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_clip: 1085, 1270, 1475  
 \\_\_draw\_backend\_closepath: .....  
 ..... 1085, 1270, 1475  
 \\_\_draw\_backend\_closestroke: ...  
 ..... 1085, 1270, 1475  
 \\_\_draw\_backend\_cm:nnnn 1201, 1214,  
 1215, 1216, 1325, 1408, 1583, 1594  
 \\_\_draw\_backend\_cm\_aux:nnnn .. 1325  
 \\_\_draw\_backend\_cm\_decompose:nnnnN  
 ..... 1331, 1360  
 \\_\_draw\_backend\_cm\_decompose\_-  
 auxi:nnnnN ..... 1360  
 \\_\_draw\_backend\_cm\_decompose\_-  
 auxii:nnnnN ..... 1360  
 \\_\_draw\_backend\_cm\_decompose\_-  
 auxiii:nnnnN ..... 1360  
 \\_\_draw\_backend\_curveto:nnnnnn ..  
 ..... 1045, 1236, 1430  
 \\_\_draw\_backend\_dash:n .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_dash\_aux:nn .. 1543  
 \\_\_draw\_backend\_dash\_pattern:nn ..  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_discardpath: ...  
 ..... 1085, 1270, 1475  
 \\_\_draw\_backend\_end: 1031, 1230, 1424  
 \\_\_draw\_backend\_evenodd\_rule: ...  
 ..... 1080, 1265, 1471  
 \\_\_draw\_backend\_fill: 1085, 1270, 1475  
 \\_\_draw\_backend\_fillstroke: .....  
 ..... 1085, 1270, 1475  
 \\_\_draw\_backend\_join\_bevel: .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_join\_miter: .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_join\_round: .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_lineto:nn .....  
 ..... 1045, 1236, 1430  
 \\_\_draw\_backend\_linewidth:n .....  
 ..... 1169, 1293, 1543  
 \\_\_draw\_backend\_literal:n .....  
 ..... 1029, 1034,  
 1038, 1042, 1044, 1047, 1055, 1063,

|                                         |                                           |
|-----------------------------------------|-------------------------------------------|
| 1072, 1086, 1089, 1090, 1091, 1092,     | \exp_last_unbraced:Nx . . . . . 462, 476  |
| 1095, 1101, 1111, 1112, 1113, 1118,     | \exp_not:N 637, 640, 2354, 2356, 2359,    |
| 1121, 1127, 1132, 1133, 1134, 1135,     | 2389, 2391, 2394, 2550, 2552, 2555,       |
| 1140, 1141, 1144, 1150, 1160, 1166,     | 2561, 2563, 2566, 2603, 2604, 2610,       |
| 1171, 1184, 1188, 1190, 1192, 1194,     | 2611, 2630, 2635, 2767, 2775, 2791        |
| 1196, 1198, 1200, 1203, 1208, 1209,     | \exp_not:n . . 48, 89, 100, 128, 1986,    |
| 1210, 1211, 1212, 1213, 1217, 1218,     | 1991, 2277, 2505, 2506, 2520, 2521,       |
| 1220, 1221, 1222, 1223, 1224, 1228,     | 2533, 2534, 2688, 2693, 2704, 2800        |
| 1238, 1243, 1248, 1258, 1271, 1273,     | \ExplBackendFileDate . . . . . 1          |
| 1275, 1278, 1283, 1288, 1292, 1295,     |                                           |
| 1308, 1312, 1314, 1316, 1318, 1320,     |                                           |
| 1322, 1324, 1422, 1482, 1501, 1527      |                                           |
| \__draw_backend_miterlimit:n . . .      | <b>F</b>                                  |
| . . . . . 1169, 1293, 1543              | file commands:                            |
| \__draw_backend_moveto:nn . . . . .     | \file_compare_timestamp:nNnTF . 1716      |
| . . . . . 1045, 1236, 1430              | \file_parse_full_name:nNNN 1703, 1726     |
| \__draw_backend_nonzero_rule: . . .     | fp commands:                              |
| . . . . . 1080, 1265, 1471              | \fp_compare:nNnTF . . . . .               |
| \__draw_backend_path:n . . . . . 1475   | . 235, 282, 288, 340, 1341, 1354, 1399    |
| \__draw_backend_rectangle:nnnn . .      | \fp_eval:n . . . . . 228, 237, 250,       |
| . . . . . 1045, 1236, 1430              | 251, 276, 293, 308, 310, 333, 342,        |
| \__draw_backend_scope:n 1472, 1474,     | 353, 354, 418, 433, 434, 969, 970,        |
| 1494, 1534, 1556, 1568, 1570, 1572,     | 971, 984, 1000, 1001, 1002, 1343,         |
| 1574, 1576, 1578, 1580, 1582, 1585      | 1348, 1349, 1356, 1366, 1367, 1368,       |
| \__draw_backend_scope_begin: . . .      | 1369, 1378, 1379, 1380, 1381, 1390,       |
| . . . . . 1041, 1231, 1234              | 1391, 1392, 1393, 2274, 2428, 2814        |
| \__draw_backend_scope_end: . . . . .    | \fp_new:N . . . . . 301, 302              |
| . . . . . 1041, 1233, 1234              | \fp_set:Nn . . . . . 281, 284             |
| \__draw_backend_stroke: . . . . .       | \fp_use:N . . . . . 287, 291, 296         |
| . . . . . 1085, 1270, 1475              | \fp_zero:N . . . . . 283                  |
| \g__draw_clip_path_int . . . . .        | \c_zero_fp 235, 282, 288, 340, 1341, 1354 |
| . . 1481, 1484, 1497, 1526, 1529, 1537  |                                           |
| \g__draw_draw_clip_bool . . 1085, 1475  | <b>G</b>                                  |
| \g__draw_draw_eor_bool . . . . .        | graphics commands:                        |
| . . . 1080, 1097, 1115, 1123, 1137,     | \graphics_bb_restore:nTF . 1658, 1872     |
| 1146, 1162, 1265, 1279, 1284, 1289      | \graphics_bb_save:n . . . . 1687, 1880    |
| \g__draw_draw_path_int . . . . . 1475   | \l_graphics_decodearray_tl . . . . .      |
| \g__draw_draw_path_tl . . . . .         | . . . . . 1635, 1636,                     |
| . . 1430, 1486, 1502, 1504, 1531, 1540  | 1646, 1666, 1670, 1671, 1748, 1780,       |
| \g__draw_path_int . . . . . 1490, 1507  | 1781, 1819, 1822, 1823, 1841, 1905        |
|                                         | \graphics_extract_bb:n . . . . .          |
|                                         | . . . . . 1743, 1750, 1900, 1907          |
| <b>E</b>                                | \l_graphics_interpolate_bool . . .        |
| \endlandscape . . . . . 2724            | . . . . . 1637, 1647, 1665, 1672,         |
| \errmessage . . . . . 38                | 1749, 1782, 1818, 1824, 1842, 1906        |
| \evensidemargin . . . . . 2129          | \l_graphics_llx_dim . . . . .             |
| exp commands:                           | . . . . . 1620, 1759, 1811, 1918          |
| \exp_after:wN . . . . . 151, 457, 1889  | \l_graphics_lly_dim . . . . .             |
| \exp_args:Ne . . . . . 682              | . . . . . 1621, 1760, 1812, 1919          |
| \exp_args:Nf . . . . . 1174, 1298, 2051 | \l_graphics_name_tl . . . . . 1721        |
| \exp_args:NNf . . . . . 228, 276, 333   | \l_graphics_page_int . . . . .            |
| \exp_args:Nnx . . . . . 2038, 2710      | . . . . . 1631, 1651, 1652, 1676,         |
| \exp_args:NV . . . . . 453              | 1677, 1741, 1778, 1779, 1805, 1806,       |
| \exp_args:Nx . . . . .                  | 1834, 1847, 1848, 1887, 1888, 1898        |
| . . 1707, 1728, 1995, 2010, 2125, 2676  | \l_graphics_pagebox_tl . . . . .          |
|                                         | . . . . . 49, 1632, 1650,                 |



[1678](#), [1679](#), [1742](#), [1776](#), [1777](#), [1807](#),  
[1809](#), [1835](#), [1856](#), [1857](#), [1889](#), [1899](#)  
`\graphics_read_bb:n` . [1614](#), [1737](#), [1895](#)  
`\l_graphics_urx_dim` .....  
. . [1622](#), [1683](#), [1761](#), [1813](#), [1878](#), [1920](#)  
`\l_graphics_ury_dim` . . [1623](#), [1684](#),  
[1762](#), [1814](#), [1879](#), [1921](#), [1929](#), [1930](#)  
graphics internal commands:  
`\l__graphics_backend_dir_str` . [1696](#)  
`\l__graphics_backend_ext_str` . [1696](#)  
`\__graphics_backend_getbb_auxi:n`  
..... [1629](#)  
`\__graphics_backend_getbb_-`  
`auxi:nN` ..... [1832](#)  
`\__graphics_backend_getbb_-`  
`auxii:n` ..... [1629](#)  
`\__graphics_backend_getbb_-`  
`auxii:nnN` ..... [1832](#)  
`\__graphics_backend_getbb_-`  
`auxiii:nNnn` ..... [1832](#)  
`\__graphics_backend_getbb_-`  
`auxiv:nnNnn` ..... [1832](#)  
`\__graphics_backend_getbb_-`  
`auxv:nNnn` ..... [1832](#)  
`\__graphics_backend_getbb_-`  
`auxvi:nNnn` ..... [1873](#), [1875](#)  
`\__graphics_backend_getbb_eps:n` .  
..... [1614](#), [1696](#), [1737](#), [1895](#)  
`\__graphics_backend_getbb_eps:nm`  
..... [1696](#)  
`\__graphics_backend_getbb_eps:nn`  
..... [1707](#), [1714](#)  
`\__graphics_backend_getbb_jpg:n` .  
..... [1629](#), [1737](#), [1832](#), [1896](#)  
`\__graphics_backend_getbb_-`  
`pagebox:w` ..... [1832](#), [1889](#)  
`\__graphics_backend_getbb_pdf:n` .  
..... [1629](#), [1722](#), [1737](#), [1832](#), [1903](#)  
`\__graphics_backend_getbb_png:n` .  
..... [1629](#), [1737](#), [1832](#), [1896](#)  
`\__graphics_backend_include:nn` [1909](#)  
`\__graphics_backend_include_-`  
`auxi:nn` ..... [1754](#)  
`\__graphics_backend_include_-`  
`auxii:nnn` ..... [1754](#)  
`\__graphics_backend_include_-`  
`auxiii:nnn` ..... [1754](#)  
`\__graphics_backend_include_-`  
`bitmap_quote:w` ..... [1883](#), [1924](#)  
`\__graphics_backend_include_-`  
`eps:n` ..... [1615](#), [1696](#), [1754](#), [1909](#)  
`\__graphics_backend_include_-`  
`jpg:n` ..... [1689](#), [1754](#), [1924](#)

`\__graphics_backend_include_-`  
`pdf:n` . . [1689](#), [1728](#), [1754](#), [1883](#), [1909](#)  
`\__graphics_backend_include_pdf_-`  
`quote:w` ..... [1886](#), [1891](#)  
`\__graphics_backend_include_-`  
`png:n` ..... [1689](#), [1754](#), [1924](#)  
`\l__graphics_backend_name_str` . [1696](#)  
`\l__graphics_graphics_attr_tl` ...  
..... [1628](#), [1633](#),  
[1640](#), [1648](#), [1658](#), [1685](#), [1687](#), [1692](#)  
`\l__graphics_internal_box` .....  
. . [1681](#), [1683](#), [1684](#), [1877](#), [1878](#), [1879](#)  
`\g__graphics_track_int` .....  
..... [1753](#), [1799](#), [1800](#)  
group commands:  
`\group_begin:` .....  
..... [143](#), [171](#), [190](#), [2283](#), [2433](#), [2820](#)  
`\group_end:` . [156](#), [179](#), [2303](#), [2447](#), [2847](#)  
`\group_insert_after:N` . [593](#), [609](#), [620](#)

## H

hbox commands:  
`\hbox:n` ..... [2057](#), [2060](#),  
[2132](#), [2138](#), [2287](#), [2291](#), [2824](#), [2833](#)  
`\hbox_overlap_right:n` ..... [223](#),  
[255](#), [271](#), [312](#), [328](#), [356](#), [440](#), [1219](#), [1414](#)  
`\hbox_set:Nn` ..... [1681](#),  
[1877](#), [2124](#), [2156](#), [2284](#), [2434](#), [2821](#)  
`\hbox_set:Nw` ..... [2107](#)  
`\hbox_set_end:` ..... [2122](#)  
`\hbox_unpack:N` ..... [2243](#)

## I

int commands:  
`\int_compare:nNnTF` .....  
..... [505](#), [521](#), [615](#), [1651](#),  
[1676](#), [1778](#), [1805](#), [1847](#), [1887](#), [2215](#),  
[2307](#), [2601](#), [2629](#), [2765](#), [2772](#), [2788](#)  
`\int_const:Nn` ..... [149](#), [155](#),  
[511](#), [546](#), [1685](#), [1800](#), [1955](#), [2479](#), [2667](#)  
`\int_eval:n` . . . . [528](#), [537](#), [566](#), [576](#),  
[678](#), [687](#), [700](#), [702](#), [706](#), [719](#), [2579](#),  
[2604](#), [2611](#), [2624](#), [2850](#), [2858](#), [2863](#)  
`\int_gincr:N` ..... [197](#), [363](#),  
[510](#), [1481](#), [1526](#), [1799](#), [1954](#), [2023](#),  
[2067](#), [2141](#), [2666](#), [2709](#), [2745](#), [2767](#)  
`\int_gset:Nn` ..... [172](#), [191](#), [2204](#)  
`\int_gset_eq:NN` [180](#), [2068](#), [2142](#), [2746](#)  
`\int_if_exist:NTF` ..... [1789](#)  
`\int_if_odd:nTF` ..... [2127](#)  
`\int_new:N` ..... [163](#), [164](#),  
[410](#), [507](#), [599](#), [1507](#), [1753](#), [1950](#),  
[2048](#), [2079](#), [2081](#), [2662](#), [2727](#), [2758](#)  
`\int_set_eq:NN` ..... [168](#), [187](#), [2216](#)

|                                                                                                                                                             |                                                                                                                  |                                                                                                                                                                               |                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\int_step_function:nnnN</code> . . . . .                                                                                                              | 704                                                                                                              | <code>\__kernel_backend_scope_begin:n</code> .                                                                                                                                |                                                                                                                                           |
| <code>\int_use:N</code> . . . . .                                                                                                                           |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">165</a> , <a href="#">385</a> , <a href="#">413</a> , <a href="#">426</a>                                                     |
| . <a href="#">365</a> , <a href="#">396</a> , <a href="#">515</a> , <a href="#">652</a> , <a href="#">788</a> , <a href="#">831</a> , <a href="#">903</a> , |                                                                                                                  | <code>\__kernel_backend_scope_end:</code> . . .                                                                                                                               |                                                                                                                                           |
| <a href="#">1484</a> , <a href="#">1490</a> , <a href="#">1497</a> , <a href="#">1529</a> , <a href="#">1537</a> , <a href="#">1652</a> ,                   |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">75</a> , <a href="#">102</a> , <a href="#">138</a> , <a href="#">165</a> , <a href="#">224</a> , <a href="#">242</a> ,        |
| <a href="#">1677</a> , <a href="#">1692</a> , <a href="#">1779</a> , <a href="#">1792</a> , <a href="#">1804</a> , <a href="#">1806</a> ,                   |                                                                                                                  | <a href="#">256</a> , <a href="#">272</a> , <a href="#">299</a> , <a href="#">313</a> , <a href="#">329</a> , <a href="#">345</a> , <a href="#">357</a> ,                     |                                                                                                                                           |
| <a href="#">1888</a> , <a href="#">1961</a> , <a href="#">2026</a> , <a href="#">2039</a> , <a href="#">2043</a> , <a href="#">2071</a> ,                   |                                                                                                                  | <a href="#">408</a> , <a href="#">422</a> , <a href="#">441</a> , <a href="#">1235</a> , <a href="#">1418</a> , <a href="#">1429</a> , <a href="#">1607</a>                   |                                                                                                                                           |
| <a href="#">2078</a> , <a href="#">2146</a> , <a href="#">2247</a> , <a href="#">2490</a> , <a href="#">2500</a> , <a href="#">2673</a> ,                   |                                                                                                                  | <code>\g__kernel_backend_scope_int</code> . . .                                                                                                                               |                                                                                                                                           |
| <a href="#">2711</a> , <a href="#">2716</a> , <a href="#">2749</a> , <a href="#">2757</a> , <a href="#">2775</a> , <a href="#">2791</a>                     |                                                                                                                  | <a href="#">163</a> , <a href="#">170</a> , <a href="#">172</a> , <a href="#">177</a> , <a href="#">181</a> , <a href="#">189</a> , <a href="#">191</a> , <a href="#">197</a> |                                                                                                                                           |
| <code>\int_value:w</code> . . . . .                                                                                                                         |                                                                                                                  | <code>\l__kernel_backend_scope_int</code> . . .                                                                                                                               |                                                                                                                                           |
| . . . . .                                                                                                                                                   | <a href="#">2354</a> , <a href="#">2389</a> , <a href="#">2550</a> , <a href="#">2561</a> , <a href="#">2579</a> | . . . . .                                                                                                                                                                     | <a href="#">163</a> , <a href="#">169</a> , <a href="#">182</a> , <a href="#">188</a>                                                     |
| <code>\int_zero:N</code> . . .                                                                                                                              | <a href="#">1631</a> , <a href="#">1741</a> , <a href="#">1834</a> , <a href="#">1898</a>                        | <code>\__kernel_color_stack_init:Nnn</code> . .                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">505</a> , <a href="#">544</a>                                                                                                 |
|                                                                                                                                                             |                                                                                                                  | <code>\l__kernel_color_stack_int</code> . . . . .                                                                                                                             |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">599</a> , <a href="#">608</a> , <a href="#">612</a>                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\__kernel_color_stack_pop:n</code> . . . . .                                                                                                                            |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">521</a> , <a href="#">558</a> , <a href="#">612</a>                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\__kernel_color_stack_push:nn</code> . . .                                                                                                                              |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">521</a> , <a href="#">558</a> , <a href="#">608</a>                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\__kernel_dependency_version-</code>                                                                                                                                    |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | check:Nn . . . . .                                                                                                                                                            | <a href="#">1</a>                                                                                                                         |
|                                                                                                                                                             |                                                                                                                  | <code>\__kernel_dependency_version-</code>                                                                                                                                    |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | check:nn . . . . .                                                                                                                                                            | <a href="#">27</a> , <a href="#">29</a>                                                                                                   |
|                                                                                                                                                             |                                                                                                                  | <code>\c__kernel_sys_dvipdfmx_version-</code>                                                                                                                                 |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | int . . . . .                                                                                                                                                                 |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . <a href="#">143</a> , <a href="#">505</a> , <a href="#">521</a> , <a href="#">615</a> , <a href="#">2765</a> , <a href="#">2772</a> , <a href="#">2788</a>                  |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  |                                                                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <b>L</b>                                                                                                                                                                      |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\landscape</code> . . . . .                                                                                                                                             | <a href="#">2720</a> , <a href="#">2722</a>                                                                                               |
|                                                                                                                                                             |                                                                                                                  |                                                                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <b>M</b>                                                                                                                                                                      |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | math commands:                                                                                                                                                                |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\c_math_toggle_token</code> . . . .                                                                                                                                     | <a href="#">2110</a> , <a href="#">2120</a>                                                                                               |
|                                                                                                                                                             |                                                                                                                  | <code>\MessageBreak</code> . . . . .                                                                                                                                          | <a href="#">40</a>                                                                                                                        |
|                                                                                                                                                             |                                                                                                                  | mode commands:                                                                                                                                                                |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\mode_if_horizontal:TF</code> . . .                                                                                                                                     | <a href="#">2206</a> , <a href="#">2213</a>                                                                                               |
|                                                                                                                                                             |                                                                                                                  | <code>\mode_if_math:TF</code> . . . . .                                                                                                                                       | <a href="#">2104</a>                                                                                                                      |
|                                                                                                                                                             |                                                                                                                  |                                                                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <b>O</b>                                                                                                                                                                      |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\oddsidemargin</code> . . . . .                                                                                                                                         | <a href="#">2128</a>                                                                                                                      |
|                                                                                                                                                             |                                                                                                                  |                                                                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <b>P</b>                                                                                                                                                                      |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | pdf commands:                                                                                                                                                                 |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_if_exist:nTF</code> . . . . .                                                                                                                               | <a href="#">843</a>                                                                                                                       |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_last:</code> . .                                                                                                                                            | <a href="#">825</a> , <a href="#">832</a> , <a href="#">897</a> , <a href="#">904</a>                                                     |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_new:nn</code> . . . . .                                                                                                                                     | <a href="#">845</a>                                                                                                                       |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_now:nn</code> . . . . .                                                                                                                                     |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">813</a> , <a href="#">835</a> , <a href="#">839</a> , <a href="#">863</a> , <a href="#">870</a> , <a href="#">907</a>         |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_ref:n</code> . . . . .                                                                                                                                      | <a href="#">858</a>                                                                                                                       |
|                                                                                                                                                             |                                                                                                                  | <code>\pdf_object_write:nn</code> . . . . .                                                                                                                                   | <a href="#">846</a>                                                                                                                       |
|                                                                                                                                                             |                                                                                                                  | pdf internal commands:                                                                                                                                                        |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | <code>\__pdf_backend:n</code> . . . . .                                                                                                                                       | <a href="#">2655</a> ,                                                                                                                    |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">2659</a> , <a href="#">2661</a> , <a href="#">2687</a> , <a href="#">2692</a> , <a href="#">2701</a> , <a href="#">2747</a> , |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">2769</a> , <a href="#">2785</a> , <a href="#">2798</a> , <a href="#">2826</a> , <a href="#">2827</a> , <a href="#">2835</a>   |
|                                                                                                                                                             |                                                                                                                  | <code>\__pdf_backend_annotation:nnnn</code> . .                                                                                                                               |                                                                                                                                           |
|                                                                                                                                                             |                                                                                                                  | . . . . .                                                                                                                                                                     | <a href="#">2049</a> , <a href="#">2339</a> , <a href="#">2728</a>                                                                        |

|                                                                                                                                                                                                 |                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>K</b>                                                                                                                                                                                        |                                                                                                                                                       |
| kernel internal commands:                                                                                                                                                                       |                                                                                                                                                       |
| <code>\__kernel_backend_align_begin:</code> . .                                                                                                                                                 |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">64</a> , <a href="#">208</a> , <a href="#">232</a> , <a href="#">247</a>                                                                  |
| <code>\__kernel_backend_align_end:</code> . . .                                                                                                                                                 |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">64</a> , <a href="#">222</a> , <a href="#">240</a> , <a href="#">254</a>                                                                  |
| <code>\g__kernel_backend_header_bool</code> . .                                                                                                                                                 |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">57</a> , <a href="#">634</a>                                                                                                              |
| <code>\__kernel_backend_literal:n</code> . . . .                                                                                                                                                |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">46</a> , <a href="#">52</a> ,                                                                                                             |
| <a href="#">55</a> , <a href="#">62</a> , <a href="#">66</a> , <a href="#">73</a> , <a href="#">76</a> , <a href="#">78</a> , <a href="#">134</a> , <a href="#">137</a> , <a href="#">139</a> , |                                                                                                                                                       |
| <a href="#">141</a> , <a href="#">161</a> , <a href="#">337</a> , <a href="#">350</a> , <a href="#">512</a> , <a href="#">525</a> , <a href="#">534</a> ,                                       |                                                                                                                                                       |
| <a href="#">588</a> , <a href="#">596</a> , <a href="#">619</a> , <a href="#">625</a> , <a href="#">648</a> , <a href="#">784</a> , <a href="#">1033</a> ,                                      |                                                                                                                                                       |
| <a href="#">1039</a> , <a href="#">1338</a> , <a href="#">1345</a> , <a href="#">1351</a> , <a href="#">1411</a> , <a href="#">1416</a> ,                                                       |                                                                                                                                                       |
| <a href="#">1617</a> , <a href="#">1756</a> , <a href="#">1791</a> , <a href="#">1801</a> , <a href="#">1915</a> , <a href="#">1926</a> ,                                                       |                                                                                                                                                       |
| <a href="#">2656</a> , <a href="#">2795</a> , <a href="#">2850</a> , <a href="#">2854</a> , <a href="#">2859</a> , <a href="#">2864</a>                                                         |                                                                                                                                                       |
| <code>\__kernel_backend_literal_page:n</code>                                                                                                                                                   |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">92</a> , <a href="#">136</a> , <a href="#">2650</a> , <a href="#">2652</a> , <a href="#">2869</a> , <a href="#">2871</a>                  |
| <code>\__kernel_backend_literal_pdf:n</code> .                                                                                                                                                  |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">81</a> , <a href="#">133</a> , <a href="#">263</a> , <a href="#">320</a> , <a href="#">919</a> ,                                          |
| <a href="#">921</a> , <a href="#">923</a> , <a href="#">925</a> , <a href="#">927</a> , <a href="#">929</a> , <a href="#">931</a> , <a href="#">933</a> , <a href="#">1228</a>                  |                                                                                                                                                       |
| <code>\__kernel_backend_literal-</code>                                                                                                                                                         |                                                                                                                                                       |
| postscript:n . . . . .                                                                                                                                                                          |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">51</a> , <a href="#">67</a> , <a href="#">68</a> , <a href="#">72</a> , <a href="#">209</a> , <a href="#">210</a> , <a href="#">212</a> , |
| <a href="#">213</a> , <a href="#">221</a> , <a href="#">233</a> , <a href="#">248</a> , <a href="#">1029</a> , <a href="#">2309</a> , <a href="#">2321</a>                                      |                                                                                                                                                       |
| <code>\__kernel_backend_literal_svg:n</code> .                                                                                                                                                  |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">160</a> , <a href="#">167</a> , <a href="#">178</a> , <a href="#">186</a> ,                                                               |
| <a href="#">196</a> , <a href="#">364</a> , <a href="#">366</a> , <a href="#">383</a> , <a href="#">1422</a> , <a href="#">1595</a> , <a href="#">1606</a>                                      |                                                                                                                                                       |
| <code>\__kernel_backend_matrix:n</code> . . . . .                                                                                                                                               |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">120</a> , <a href="#">285</a> , <a href="#">306</a> , <a href="#">1328</a>                                                                |
| <code>\__kernel_backend_postscript:n</code> . .                                                                                                                                                 |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">54</a> , <a href="#">590</a> , <a href="#">591</a> , <a href="#">939</a> ,                                                                |
| <a href="#">941</a> , <a href="#">943</a> , <a href="#">945</a> , <a href="#">947</a> , <a href="#">949</a> , <a href="#">951</a> , <a href="#">953</a> ,                                       |                                                                                                                                                       |
| <a href="#">1944</a> , <a href="#">2000</a> , <a href="#">2015</a> , <a href="#">2057</a> , <a href="#">2063</a> , <a href="#">2100</a> ,                                                       |                                                                                                                                                       |
| <a href="#">2132</a> , <a href="#">2139</a> , <a href="#">2143</a> , <a href="#">2157</a> , <a href="#">2185</a> , <a href="#">2229</a> ,                                                       |                                                                                                                                                       |
| <a href="#">2236</a> , <a href="#">2242</a> , <a href="#">2250</a> , <a href="#">2257</a> , <a href="#">2287</a> , <a href="#">2291</a>                                                         |                                                                                                                                                       |
| <code>\__kernel_backend_scope:n</code> . . . . .                                                                                                                                                |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">165</a> , <a href="#">393</a> , <a href="#">398</a> , <a href="#">1007</a> , <a href="#">1427</a>                                         |
| <code>\__kernel_backend_scope_begin:</code> . .                                                                                                                                                 |                                                                                                                                                       |
| . . . . .                                                                                                                                                                                       | <a href="#">75</a> , <a href="#">102</a> , <a href="#">138</a> ,                                                                                      |
| <a href="#">165</a> , <a href="#">207</a> , <a href="#">231</a> , <a href="#">246</a> , <a href="#">262</a> , <a href="#">279</a> , <a href="#">305</a> ,                                       |                                                                                                                                                       |
| <a href="#">319</a> , <a href="#">336</a> , <a href="#">349</a> , <a href="#">1234</a> , <a href="#">1406</a> , <a href="#">1426</a> , <a href="#">1593</a>                                     |                                                                                                                                                       |

|                                  |                                                                                                                              |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| \_pdf_backend_annotation_-       |                                                                                                                              |
| aux:nnnn                         | 2051, 2054, 2728                                                                                                             |
| \g_pdf_backend_annotation_int    | 2048, 2068, 2078, 2727, 2746, 2757                                                                                           |
| \_pdf_backend_annotation_last:   | 2077, 2352, 2756                                                                                                             |
| \_pdf_backend_bdc:nn             | 2333, 2649, 2868, 2890                                                                                                       |
| \_pdf_backend_catalog_gput:nn    | 1946, 2449, 2658, 2874                                                                                                       |
| \_pdf_backend_compress_objects:n | 2305, 2570, 2849, 2884                                                                                                       |
| \_pdf_backend_compresslevel:n    | 2305, 2570, 2849, 2884                                                                                                       |
| \l_pdf_backend_content_box       | 2046, 2107, 2131, 2134, 2136, 2165, 2176                                                                                     |
| \_pdf_backend_destination:nn     | 2255, 2408, 2796                                                                                                             |
| \_pdf_backend_destination_-      |                                                                                                                              |
| box:nn                           | 2255, 2408, 2796                                                                                                             |
| \_pdf_backend_emc:               | 2333, 2649, 2868, 2890                                                                                                       |
| \_pdf_backend_info_gput:nn       | 1946, 2449, 2658, 2874                                                                                                       |
| \_pdf_backend_link:nw            | 2088                                                                                                                         |
| \_pdf_backend_link_aux:nw        | 2088                                                                                                                         |
| \_pdf_backend_link_begin:n       | 2759                                                                                                                         |
| \_pdf_backend_link_begin:nnw     | 2363                                                                                                                         |
| \_pdf_backend_link_begin:nw      | 2089, 2091, 2092                                                                                                             |
| \_pdf_backend_link_begin_aux:nw  | 2095, 2097                                                                                                                   |
| \_pdf_backend_link_begin_-       |                                                                                                                              |
| goto:nnw                         | 2088, 2363, 2759                                                                                                             |
| \_pdf_backend_link_begin_-       |                                                                                                                              |
| user:nnw                         | 2088, 2363, 2759                                                                                                             |
| \g_pdf_backend_link_bool         | 2083, 2094, 2099, 2114, 2152                                                                                                 |
| \g_pdf_backend_link_dict_tl      | 2080, 2102, 2147                                                                                                             |
| \_pdf_backend_link_end:          | 2088, 2363, 2759                                                                                                             |
| \_pdf_backend_link_end_aux:      | 2088                                                                                                                         |
| \g_pdf_backend_link_int          | 2079, 2142, 2146, 2247, 2758, 2767, 2775, 2791                                                                               |
| \_pdf_backend_link_last:         | 2246, 2387, 2786                                                                                                             |
| \_pdf_backend_link_margin:n      | 2248, 2398, 2794                                                                                                             |
| \g_pdf_backend_link_math_bool    | 2082, 2105, 2106, 2109, 2119                                                                                                 |
| \_pdf_backend_link_minima:       | 2088                                                                                                                         |
| \_pdf_backend_link_outerbox:n    | 2088                                                                                                                         |
| \g_pdf_backend_link_sf_int       | 2081, 2204, 2215, 2216                                                                                                       |
| \_pdf_backend_link_sf_restore:   | 2088                                                                                                                         |
| \_pdf_backend_link_sf_save:      | 2088                                                                                                                         |
| \l_pdf_backend_model_box         | 2047, 2124, 2156, 2164, 2175, 2190, 2192                                                                                     |
| \_pdf_backend_objcompresslevel:n | 2570                                                                                                                         |
| \g_pdf_backend_object_int        | 1950, 1954, 1957, 2023, 2026, 2039, 2043, 2067, 2068, 2071, 2141, 2142, 2662, 2666, 2669, 2709, 2711, 2716, 2745, 2746, 2749 |
| \_pdf_backend_object_last:       | 2042, 2548, 2715, 2876                                                                                                       |
| \_pdf_backend_object_new:nn      | 1952, 2470, 2664, 2876                                                                                                       |
| \_pdf_backend_object_now:nn      | 2021, 2522, 2707, 2876                                                                                                       |
| \g_pdf_backend_object_prop       | 1950, 1958, 1969, 1979, 2469, 2487, 2503, 2662, 2670, 2677                                                                   |
| \_pdf_backend_object_ref:n       | 1952, 1966, 1980, 2470, 2664, 2683, 2876                                                                                     |
| \_pdf_backend_object_write:nn    | 1962, 2491, 2674, 2876                                                                                                       |
| \_pdf_backend_object_write:nnn   | 2674                                                                                                                         |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| array:nn                         | 1962, 2674                                                                                                                   |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| dict:nn                          | 1962, 2674                                                                                                                   |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| fstream:nn                       | 1962, 2674                                                                                                                   |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| fstream:nnn                      | 1996, 1998                                                                                                                   |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| stream:nn                        | 1962, 2674                                                                                                                   |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| stream:nnn                       | 1962                                                                                                                         |
| \_pdf_backend_object_write_-     |                                                                                                                              |
| stream:nnnn                      | 2674                                                                                                                         |
| \_pdf_backend_pageobject_ref:n   | 2044, 2559, 2717, 2876                                                                                                       |
| \_pdf_backend_pdfmark:n          | 1943, 1947, 1949, 1964, 1985, 1990, 2024, 2069, 2258, 2292, 2334, 2336                                                       |
| \_pdf_backend_version_major:     | 2331, 2626, 2858, 2859, 2866, 2888                                                                                           |
| \_pdf_backend_version_major_-    |                                                                                                                              |
| gset:n                           | 2329, 2598, 2856, 2886                                                                                                       |
| \_pdf_backend_version_minor:     | 2331, 2626, 2863, 2864, 2866, 2888                                                                                           |
| \_pdf_backend_version_minor_-    |                                                                                                                              |
| gset:n                           | 2329, 2598, 2856, 2886                                                                                                       |



## T

TeX and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> commands:

\ccclv ..... 2225, 2227, 2235  
 \@makecol@hook ..... 2219  
 \current@color . 14, 453, 457, 463, 477  
 \special ..... 2

tex commands:

\tex\_baselineskip:D ..... 2196  
 \tex\_endinput:D ..... 44  
 \tex\_global:D .....  
 ..... 2572, 2589, 2603, 2610, 2617  
 \tex\_immediate:D .....  
 ..... 1663, 2494, 2497, 2525, 2528  
 \tex\_kern:D ..... 2062, 2064  
 \tex luatexversion:D .... 2601, 2629  
 \tex\_pdfannot:D ..... 2345  
 \tex\_pdfcatalog:D ..... 2455  
 \tex\_pdfcolorstack:D ..... 564, 574  
 \tex\_pdfcolorstackinit:D ..... 552  
 \tex\_pdfcompresslevel:D ..... 2577  
 \tex\_pdfdest:D ..... 2414, 2439  
 \tex\_pdfendlink:D ..... 2384  
 \tex\_pdfextension:D .....  
 ..... 84, 95, 105, 114, 123,  
 561, 571, 2342, 2370, 2381, 2411,  
 2436, 2452, 2462, 2473, 2494, 2525  
 \tex\_pdffeedback:D .....  
 ... 549, 2356, 2391, 2482, 2552, 2563  
 \tex\_pdfinfo:D ..... 2465  
 \tex\_pdflastannot:D ..... 2359  
 \tex\_pdflastlink:D ..... 2394  
 \tex\_pdflastobj:D ..... 2485, 2555  
 \tex\_pdflastximage:D .... 1682, 1686  
 \tex\_pdflinkmargin:D ..... 2404  
 \tex\_pdfliteral:D ..... 87, 98  
 \tex\_pdfmajorversion:D .....  
 ..... 2608, 2610, 2634, 2635  
 \tex\_pdfminorversion:D ... 2622, 2646  
 \tex\_pdfobj:D ..... 2476, 2497, 2528  
 \tex\_pdfobjcompresslevel:D ... 2594  
 \tex\_pdfpageref:D ..... 2566  
 \tex\_pdfrefximage:D ..... 1682, 1691  
 \tex\_pdfrestore:D ..... 117  
 \tex\_pdfsave:D ..... 108  
 \tex\_pdfsetmatrix:D ..... 126  
 \tex\_pdfstartlink:D ..... 2373  
 \tex\_pdfvariable:D ..... 2401,  
 2574, 2591, 2603, 2619, 2630, 2643

\tex\_pdfximage:D ..... 1663  
 \tex\_spacefactor:D ..... 2207, 2216  
 \tex\_special:D ..... 46  
 \tex\_the:D .... 1686, 2630, 2635, 2641  
 \tex\_XeTeXpdffile:D ..... 1843, 1885  
 \tex\_XeTeXpicfile:D ..... 1836  
 TeXcolorseparation ..... 2897  
 \textwidth ..... 2191

tl commands:

\c\_space\_tl ..... 287, 292, 295,  
 515, 743, 1466, 1619, 1620, 1621,  
 1622, 1758, 1759, 1760, 1761, 1806,  
 1809, 1811, 1812, 1813, 1814, 1886,  
 1888, 1917, 1918, 1919, 1920, 2147,  
 2361, 2396, 2557, 2568, 2749, 2776  
 \tl\_clear:N ..... 1632, 1640, 1646,  
 1742, 1748, 1835, 1841, 1899, 1905  
 \tl\_gclear:N ..... 1504, 1540  
 \tl\_gset:Nn ..... 1463, 2102  
 \tl\_if\_blank:nTF .....  
 516, 554, 691, 708, 715, 733, 817, 912  
 \tl\_if\_empty:nTF . 1466, 1635, 1670,  
 1678, 1776, 1780, 1807, 1822, 1856  
 \tl\_if\_empty:nTF ..... 1560  
 \tl\_if\_empty\_p:N ..... 1666, 1819  
 \tl\_if\_head\_is\_space:nTF ..... 453  
 \tl\_new:N ..... 1470, 1628, 2080, 2084  
 \tl\_put\_left:Nn ..... 2724  
 \tl\_put\_right:Nn ..... 2223, 2722  
 \tl\_set:Nn . 455, 467, 483, 486, 489,  
 493, 496, 1633, 1648, 1721, 2085, 2241  
 \tl\_to\_str:n ..... 1956,  
 1961, 2480, 2490, 2501, 2668, 2673  
 \tl\_use:N ..... 775, 851

## U

use commands:

\use:N ..... 43, 1978, 2038, 2682, 2710  
 \use:n ..... 61, 457, 493,  
 638, 827, 899, 965, 980, 996, 1174,  
 1298, 1363, 1375, 1387, 1545, 1863  
 \use\_none:n ..... 1560, 1562, 2219

## V

\value ..... 2127  
 vbox commands:  
 \vbox:n ..... 2734  
 \vbox\_set:Nn ..... 2227  
 \vbox\_unpack\_drop:N ..... 2235